零起点学自动化技术丛书

零起点学Linux系统管理

LINGQIDIAN XUE LINUX XITONG GUANLI

范海绍　李方园　等编著



机械工业出版社
CHINA MACHINE PRESS

含1DVD



VISIT…

LANZAROTE Caliente.com

零起点学自动化技术丛书

零起点学 Linux 系统管理

范海绍　李方园　等编著



机 械 工 业 出 版 社

本书以当前流行的 Enterprise Red Hat Linux 4.0 为蓝本，全面地介绍了 Linux 操作系统的使用及配置。全书共分为 10 讲，内容涉及 Linux 系统的安装、基本网络配置、用户管理、文件系统管理、文件编辑、系统管理、软件包安装、基本网络功能应用和扩展内容 Shell 程序设计基础、C 语言编程工具等。

本书以企业实际项目场景为中心展开教学，同时提出完整的教学目标、实验指导、项目完成步骤详解和课后练习，内容详尽、精练，使学生对学习 Linux 系统有明确的目标，并能很快地入门，掌握基本的应用配置方法。

本书充分考虑到教师在授课中所需要的材料，完整地综合了项目、大纲、实验、练习和技能训练，并在随书的光盘中提供电子课件、课程技能训练与考核记录本、上机考核试题、期末考核试题和考核软件、虚拟机软件和 Enterprise Red Hat Linux 4.0 虚拟光盘。

本书不仅可以作为高等院校相关专业的授课教材，还可以作为 Linux 的培训教材，同时也可供广大 Linux 的爱好者自学使用。

图书在版编目（CIP）数据

零起点学 Linux 系统管理/范海绍等编著. —北京：机械工业出版社，2012. 2

（零起点学自动化技术丛书）

ISBN 978-7-111-37309-4

Ⅰ. ①零…　Ⅱ. ①范…　Ⅲ. ①Linux 操作系统　Ⅳ. ①TP316. 89

中国版本图书馆 CIP 数据核字（2012）第 014639 号

机械工业出版社（北京市百万庄大街 22 号　邮政编码 100037）

策划编辑：林春泉　责任编辑：赵　任

版式设计：石　冉　责任校对：王　欣

封面设计：路恩中　责任印制：乔　宇

三河市宏达印刷有限公司印刷

2012 年 3 月第 1 版第 1 次印刷

184mm×260mm · 10.25 印张 · 246 千字

0001—3000册

标准书号：ISBN 978-7-111-37309-4

ISBN 978-7-89433-316-2（光盘）

定价：29.80 元（含 1DVD）

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务

社服务中心：（010）88361066

销售一部：（010）68326294

销售二部：（010）88379649

读者购书热线：（010）88379203

网络服务

门户网：http：//www. cmpbook. com

教材网：http：//www. cmpedu. com

封面无防伪标均为盗版

前　　言

Linux 应用已从前几年的盲目推进和前一段的低迷期摆脱出来，更具理性、更重实效的 Linux 应用时代即将到来。在世界发达国家，Linux 早已涉足政府办公、军事战略以及商业运作的方方面面。在我国，随着国民经济与社会信息化的进一步深入，Linux 的应用在电子政务、电子商务等各个领域中也突显其不凡之处。Linux 在我国起步较晚，随着 Linux 在各行各业的广泛成功应用，企业对 Linux 人才的需求也将持续扩大。巨大的人才需求，使人们感觉到了学习 Linux 的迫切性，也使一些商家看到了商机。Linux 这个免费的开放源代码的操作系统正以狂风暴雨之势席卷整个世界，它不仅出现在企业服务器和专业怪才的讨论中，也开始在家用 PC 上生根。Linux 的普及程度正在与日俱增。

另外，随着计算技术、通信技术的飞速发展，特别是互联网的迅速普及和 3C（计算机、通信、消费电子）合一的加速，微型化和专业化成为发展的新趋势，嵌入式产品成为信息产业的主流。Linux 从 1991 年问世到现在，短短的十几年时间已经发展成为功能强大、设计完善的操作系统之一；可运行在 x86、Alpha、Sparc、MIPS、PPC、Motorola、NEC、ARM 等多种硬件平台，而且开放源代码，可以定制；可与各种传统的商业操作系统分庭抗争。越来越多的企业和研发机构都转向嵌入式 Linux 的开发和研究上，在新兴的嵌入式操作系统领域内也获得了飞速发展。

嵌入式系统是以应用为中心，以计算机为基础，软硬件可裁剪，适用于系统对功能、可靠性、成本、功耗严格要求的专用计算机系统。实时性是嵌入式系统的基本要求，其次，还要求代码小、速度快、可靠性高。嵌入式 Linux（Embedded Linux）是指对 Linux 经过裁剪小型化后，可固化在存储器或单片机中，应用于特定嵌入式场合的专用 Linux 操作系统。嵌入式 Linux 的开发和研究已经成为目前操作系统领域的一个热点。与其他嵌入式操作系统相比，Linux 的特点如下：

第一，Linux 系统是层次结构且内核完全开放。Linux 是由很多体积小且性能高的微内核系统组成。在内核代码完全开放的前提下，不同领域和不同层次的用户可以根据自己的应用需求方便地对内核进行改造，低成本地设计和开发出满足自己需求的嵌入式系统。

第二，强大的网络支持功能。Linux 诞生于因特网时代并具有 UNIX 的特性，保证了它支持所有标准因特网协议，并且可以利用 Linux 的网络协议栈将其开发成为嵌入式的 TCP/IP 网络协议栈。此外，Linux 还支持 ext2、fat16、fat32、romfs 等文件系统，为开发嵌入式系统应用打下了很好的基础。

第三，Linux 具备一整套工具链，容易自行建立嵌入式系统的开发环境和交叉运行环境，可以跨越嵌入式系统开发中仿真工具的障碍。Linux 也符合 IEEE POSIX. 1 标准，使应用程序具有较好的可移植性。

传统的嵌入式开发的程序调试和调试工具是用在线仿真器（ICE）实现的。它通过

取代目标板的微处理器，给目标程序提供一个完整的仿真环境，完成监视和调试程序；但一般价格比较昂贵，只适合做非常底层的调试。使用嵌入式 Linux，一旦软硬件能够支持正常的串口功能，即使不用仿真器，也可以很好地进行开发和调试工作，从而节省了一笔不小的开发费用。嵌入式 Linux 为开发者提供了一套完整的工具链（tool chain）。它利用 GNU 的 gcc 作为编译器，用 gdb、kgdb、xgdb 作为调试工具，能够很方便地实现从操作系统到应用软件各个级别的调试。

第四，Linux 具有广泛的硬件支持特性。无论是 RISC 还是 CISC、32 位还是 64 位等各种处理器，Linux 都能运行。Linux 通常使用的微处理器是 Intel x86 芯片家族，但它同样能运行于 Motorola 公司的 68K 系列 CPU 和 IBM、Apple、Motorola 公司的 PowerPC CPU 以及 Intel 公司的 StrongARM CPU 等处理器。Linux 支持各种主流硬件设备和最新硬件技术，甚至可以在没有存储管理单元（MMU）的处理器上运行。这意味着嵌入式 Linux 将具有更广泛的应用前景。

因此，笔者认为，Linux 操作系统不仅是计算机专业中网络专业、软件专业学生的必修课，也是电子专业，特别是嵌入式系统设计专业学生的必修课。在本书的安排中，我们强调字符命令的学习，而非图形界面的学习。这是因为我们的目标是培养 Linux 服务器系统管理人才或嵌入式系统开发应用人才，而不是一般的桌面应用。对于通用操作系统的桌面应用，Linux 系统还不具备与 Windows 系统对抗的优势。

本教材是编者几年来的教学经验的积累，每讲都有企业项目场景、教学目标和实验指导，这 3 部分是每讲的核心。技术环节的详细说明和解释放在每讲的引言中和最后的附录中。企业场景的设计是根据实际的需求编制的，并非一成不变，教师可根据自己的经验修改，这里只是起到抛砖引玉的作用。

参加编写的人员还有杨帆、乐斌、钟晓强、邵振翔、孙雄英、吴冬燕等。

在写作过程中，参考了国内外许多专家、学者发表的论文和著作等资料，在此表示衷心的感谢！

编　者

2011 年 12 月

目　　录

前言
第 1 讲　Linux 系统安装 …… 1
1.1　引言 …… 1
1.1.1　Linux 简介 …… 1
1.1.2　Linux 版本说明 …… 4
1.1.3　各种安装方法介绍 …… 5
1.2　企业项目场景 1：Linux 系统安装 …… 6
1.3　Linux 系统安装教学目标（4 学时） …… 6
1.4　项目一　实验指导 …… 7
1.4.1　新建虚拟机 …… 7
1.4.2　在虚拟机上安装 Linux 系统 …… 7
1.5　项目一完成步骤详解 …… 9
1.5.1　新建虚拟机 …… 9
1.5.2　在虚拟机上安装 Linux 系统 …… 12
1.6　思考题 …… 22
第 2 讲　基本网络配置 …… 23
2.1　引言 …… 23
2.1.1　网络的定义 …… 23
2.1.2　网络组成 …… 24
2.1.3　网络 IP 地址 …… 24
2.1.4　网络路由 …… 25
2.1.5　网络域名 …… 25
2.2　企业项目场景 2：基本网络配置 …… 26
2.3　Linux 系统基本网络配置教学目标（8 学时） …… 26
2.4　项目二　实验指导 …… 27
2.4.1　配置网卡 …… 27
2.4.2　测试网络 …… 27
2.4.3　用“ping”命令测试 …… 27
2.4.4　用 ifconfig 命令 …… 28
2.4.5　开通和关闭网络设备（网卡） …… 28
2.4.6　设置临时 IP 地址 …… 28
2.4.7　设置默认网关 …… 28
2.4.8　设置指定网关 …… 29
2.4.9　文件上传和下载 …… 29

2.4.10 远程登录 …… 29
2.5 项目二完成步骤详解 …… 29
2.5.1 网卡设置 …… 29
2.5.2 设置指定网关 …… 33
2.5.3 简单域名设置 …… 34
2.5.4 网络测试 …… 36
2.6 思考题 …… 38
第3讲 Linux 系统用户管理 …… 39
3.1 引言 …… 39
3.2 企业项目场景3：Linux 系统用户管理 …… 49
3.3 Linux 系统用户管理教学目标（8学时） …… 49
3.4 项目三 实验指导 …… 50
3.4.1 新建组 …… 50
3.4.2 删除组 …… 50
3.4.3 新建用户 …… 50
3.4.4 删除用户 …… 51
3.4.5 修改用户属性 …… 51
3.4.6 暂停和恢复用户账号 …… 51
3.4.7 超级用户管理（学习 Vi 编辑后再做） …… 51
3.5 项目三完成步骤详解 …… 52
3.5.1 新建组 …… 52
3.5.2 新建用户 …… 52
3.5.3 暂停和恢复用户账号 …… 54
3.5.4 超级用户管理（学习 Vi 编辑后再做） …… 54
3.6 思考题 …… 58
第4讲 Linux 文件系统管理 …… 59
4.1 引言 …… 59
4.1.1 文件和目录概述 …… 59
4.1.2 文件类型 …… 60
4.1.3 文件系统和标准文件系统布局 …… 61
4.2 企业项目场景4：Linux 文件系统管理 …… 64
4.3 Linux 文件系统管理教学目标（16学时） …… 65
4.4 项目四 实验指导 …… 65
4.4.1 文件处理 …… 65
4.4.2 目录（文件夹）处理 …… 66
4.4.3 文件和目录的存取权限 …… 67
4.4.4 文件和文件内容查找 …… 68
4.4.5 文件压缩与解压缩 …… 68
4.5 项目四完成步骤详解 …… 69

4.5.1　创建目录和文件 …… 69
4.5.2　设置文件/目录权限 …… 70
4.5.3　查找文件和文件内容 …… 73
4.5.4　文件压缩与备份 …… 73
4.6　思考题 …… 74
第 5 讲　Linux 文件编辑 …… 75
5.1　引言 …… 75
5.1.1　Vi 简介 …… 75
5.1.2　Vi 使用方法 …… 75
5.2　企业项目场景 5：Linux 文件编辑 …… 79
5.3　Linux 文件编辑教学目标（4 学时） …… 79
5.4　项目五　实验指导 …… 80
5.4.1　文件录入命令 …… 80
5.4.2　文件修改 …… 80
5.4.3　文件保存 …… 80
5.4.4　文件查找和替换 …… 81
5.4.5　其他编辑手段 …… 81
5.5　项目五完成步骤详解 …… 81
5.5.1　新建文件并打开输入区 …… 81
5.5.2　录入文件内容 …… 82
5.5.3　保存文件 …… 82
5.6　思考题 …… 84
第 6 讲　Linux 系统管理 …… 85
6.1　引言 …… 85
6.1.1　什么是磁盘配额（Quota） …… 85
6.1.2　Quota 的一般用途 …… 86
6.1.3　Quota 的使用限制 …… 86
6.1.4　Quota 的规范设置选项 …… 86
6.1.5　Linux 工作调度种类：at，crontab …… 87
6.1.6　at 运行方式 …… 87
6.1.7　cron 运行方式 …… 88
6.2　企业项目场景 6：Linux 系统管理 …… 89
6.3　Linux 系统管理教学目标（8 学时） …… 89
6.4　项目六　实验指导 …… 90
6.4.1　磁盘配额 …… 90
6.4.2　定时执行命令 …… 90
6.4.3　周期执行命令 …… 91
6.5　项目六完成步骤详解 …… 91
6.5.1　为部门和用户分配磁盘空间 …… 91

6.5.2 定时执行任务 …… 94
6.6 思考题 …… 96
第 7 讲 Linux 软件包安装 …… 97
7.1 引言 …… 97
7.1.1 RPM 软件包管理 …… 97
7.1.2 Tarball 源代码软件包 …… 99
7.2 企业项目场景 7：Linux 系统软件包安装 …… 100
7.3 Linux 系统软件包安装教学目标（4 学时） …… 100
7.4 项目七 实验指导 …… 101
7.4.1 RPM 软件包管理工具 …… 101
7.4.2 源代码软件包安装 …… 102
7.5 项目七完成步骤详解 …… 102
7.5.1 挂接光盘 …… 102
7.5.2 安装 vsftpd 软件包 …… 102
7.5.3 安装 tar 格式的源码软件包 ruby …… 103
7.6 思考题 …… 104
第 8 讲 基本网络功能应用 …… 105
8.1 引言 …… 105
8.1.1 NFS 网络概述 …… 105
8.1.2 NFS 的优点 …… 105
8.1.3 NFS 工作原理 …… 106
8.1.4 Web 服务概述 …… 107
8.1.5 Apache 服务器 …… 107
8.2 企业项目场景 8：基本网络功能应用 …… 108
8.3 基本网络功能应用教学目标（8 学时） …… 108
8.4 项目八实验指导 …… 109
8.4.1 NFS 网络 …… 109
8.4.2 Web 网络 …… 109
8.5 项目八完成步骤详解 …… 110
8.5.1 NFS 网络实现 …… 110
8.5.2 Web 网络实现 …… 113
8.6 思考题 …… 115
第 9 讲 Shell 程序设计基础 …… 116
9.1 引言 …… 116
9.2 Shell 程序设计基础教学目标（8 学时） …… 117
9.3 Shell 程序设计基础 实验指导 …… 118
9.3.1 Linux 变量 …… 118
9.3.2 重定向和管道 …… 119
9.3.3 命令替换 …… 119

9. 3. 4 算术运算 …… 119
9. 3. 5 Shell 程序执行和程序参数 …… 119
9. 3. 6 条件表达式 …… 120
9. 3. 7 控制结构 …… 121
9. 3. 8 函数 …… 124
9. 3. 9 菜单命令 select …… 125
9. 3. 10 作业控制 …… 125
第 10 讲 C 语言编程工具 …… 127
10. 1 引言 …… 127
10. 2 C 语言编程工具教学目标（4 学时） …… 127
10. 3 C 语言编程工具 实验指导 …… 128
10. 3. 1 C 编译器 …… 128
10. 3. 2 单模块程序 …… 128
10. 3. 3 多模块程序 …… 129
10. 3. 4 归档模块 ar …… 129
10. 3. 5 编译管理工具 make …… 130
10. 3. 6 make 工具与 ar 归档工具联合应用 …… 131
10. 3. 7 调试器 gdb …… 132
附录 常用命令索引及详解 …… 134
参考文献 …… 151

第1讲 Linux 系统安装

1.1 引言

1.1.1 Linux 简介

Linux 操作系统是一款优秀的操作系统，支持多用户、多线程、多进程，实时性好，功能强大且稳定。同时，它又具有良好的兼容性和可移植性，被广泛应用于各种计算机平台上。下面介绍 Linux 的产生背景、组成和特性。

1. Linux 的产生背景

对于 Linux 操作系统的产生，可以追溯到另一个操作系统 UNIX。与 Linux 相同，UNIX 也是一款相当流行的计算机操作系统，该操作系统最初是由美国贝尔实验室的 Ken Thompson、Dennis Ritchie 和其他人共同开发的。UNIX 是一个实时操作系统，可允许多人同时访问计算机，与此同时每个人可运行多个应用程序，即通常所说的多用户、多任务操作系统，该操作系统最初是为了运行于大型计算机和小型计算机上而设计的。

UNIX 操作系统以其优越的性能在工作站或小型计算机上发挥着重要作用，一直以来，该操作系统是一种大型而且要求较高的操作系统，许多种版本的 UNIX 操作系统都是为工作站环境而设计的。但随着个人计算机的日益普及，并且个人计算机的性能也在不断提高，人们也开始从事 UNIX 操作系统的个人计算机版本的开发，使 UNIX 能够在个人计算机上运行成为可能，这也是 Linux 流行起来的原因。

Linux 的前身是芬兰赫尔辛基大学一位名叫 Linus Torvalds 计算机科学系学生的个人项目。他将 Linux 建立在一个基于 PC 上运行的、名为 Minix（Minix 是由一位名为 Andrew Tannebaum 的计算机教授编写的操作系统示例程序）的操作系统之上。Minix 突出体现了 UNIX 的各种特性，后来 Minix 通过 Internet 广泛传播。Linus 的初衷是为 Minix 用户开发一种高效率的 PC UNIX 版本，称其为 Linux，并于 1991 年底首次公布于众，同年 11 月发布了 0.10 版本，12 月发布了 0.11 版本。Linux 允许免费自由地运用该系统源代码，并且鼓励其他人进一步对其进行开发。如此一来，通过 Internet 在世界范围内形成了 Linux 研究热潮，并且在不断持续着。

2. Linux 组成

Linux 操作系统的组成一般分为 3 个部分：内核（Kernel）、命令解释层（Shell 或其他操作环境）、文件结构（File Structure）。其中内核是整个操作系统的内核部分；Shell 是用户

与计算机交流的接口；文件结构是存放在存储设备上文件的组织方法。

（1）内核

内核是 Linux 系统的心脏，是运行程序和管理硬件设备的内核程序，决定着系统的性能和稳定性。内核以独占的方式执行最底层任务，保证系统正常运行，协调多个并发进程，管理进程使用的内存，使它们相互之间不产生冲突，满足进程访问磁盘的请求等。它从用户那里接受命令并把命令送给内核去执行。

Linux 内核包括几个重要部分：进程管理、内存管理、硬件设备驱动、文件系统驱动、网络管理。进程管理产生进程，以切换运行时的活动进程来实现多任务；内存管理负责分配进程的存储区域和对换空间区域、内核的部件及 buffer cache；在最底层，内核对它支持的每种硬件包含一个硬件设备驱动。因为，现实世界中存在大量不同的硬件，因此硬件设备的驱动数量很大。每个类的每个成员都有相同的与内核其他部分的接口，但具体实现是不同的，例如所有的硬盘驱动与内核其他部分接口相同，即都有初始化驱动器、读 N 扇区和写 N 扇区。内核自己提供的有些软件服务有类似的抽象属性，因此可以抽象分类。例如不同的网络协议已经被抽象为一个编程接口：BSD socket 库。另一个例子是虚拟文件系统 virtual file system（VFS）层，它从文件系统操作实现中抽象出来文件系统。每个文件系统类型提供了每个文件系统操作的实现。当一些实体企图使用一个文件系统时，请求通过 VFS 送出，它将请求发送到适当的文件系统驱动。网络管理提供了对网络标准存取和各种网络硬件的支持，它又可分为网络协议和网络驱动程序。其中，网络协议部分负责实现每一种可能的网络传输协议，而网络驱动程序负责与硬件通信。

（2）Linux Shell

Shell 是系统的用户界面，提供了用户与内核进行交互操作的一种接口，它接收用户输入的命令并把命令送入内核。操作环境在操作系统内核与用户之间提供操作界面，它实际上是一个解释器。操作系统对用户输入的命令进行解释，再将其发送到内核。Linux 存在 3 种操作环境，分别是桌面（desktop）、窗口管理器（window manager）和命令行 Shell（command line Shell）。Linux 系统中的每个用户都可以拥有自己的用户操作界面，根据自己的要求进行定制。

Shell 是一个命令解释器，它解释由用户输入的命令，并把它们送到内核。不仅如此，Shell 有自己的编程语言，用于对命令的编辑，它允许用户编写由 Shell 命令组成的程序。同 Linux 本身一样，Shell 也有多种不同的版本，目前主要的 Shell 版本如下：

1）Bourne shell　是贝尔实验室开发的；

2）BASH　是 GNU 的 Bourne Again Shell，是 GNU 操作系统上默认的 Shell；

3）Korn Shell　是对 Bourne Shell 的发展，在大部分情况下与 Bourne Shell 兼容；

4）C Shell　是 SUN 公司 Shell 的 BSD 版本。

Shell 中的命令分为内部命令和外部命令。内部命令包含在 Shell 自身之中，如 cd、exit 等，查看内部命令的方法可用 help 命令；外部命令是存在于文件系统某个目录下的具体的可执行程序，如 cp 等，查看外部命令的路径可用 which。

（3）文件结构

文件结构是存放在磁盘等存储设备上的文件的组织方法，主要体现在对文件和目录的组织上。目录提供了管理文件的一个方便而有效的途径，用户能够从一个目录切换到另一个目

录，而且可以设置目录和文件的权限、文件的共享程度。Linux 目录采用多级树形结构，用户可以浏览整个系统，进入任何一个已授权进入的目录，并访问那里的文件。

文件结构使得相互关联性使用共享数据变得容易，几个用户可以访问同一个文件。Linux 是一个多用户系统，操作系统本身的驻留程序存放在以根目录开始的专用目录中，有时被指定为系统目录。此外，用户可以创建自己的子目录，保存自己的文件，可以很容易地把文件从一个子目录移动到另一个子目录中。

内核、Shell 和文件结构一起形成了基本的操作系统结构，它们使得用户可以运行程序、管理文件以及使用系统。此外，Linux 操作系统还有许多被称为实用工具的程序，辅助用户完成一些特定的任务。

3. Linux 特性

Linux 操作系统在短时间内得到迅猛的发展，这与该操作系统良好的特性是分不开的。Linux 包含了 UNIX 操作系统的全部功能和特性。简单地说，Linux 具有 UNIX 的所有特性并且具有自己独特的魅力，主要表现在以下几个方面：

（1）开放性

开放性是指系统遵循世界标准规范，特别是遵循开放系统互联（OSI）国际标准。凡遵循国际标准所开发的硬件和软件，都能彼此兼容，可方便地实现互联。

（2）多用户

多用户是指系统资源可以被不同的用户各自拥有并使用，即使每个用户对自己的资源（如文件、设备）有特定权限，也互不影响，Linux 和 UNIX 都具有多用户特性。

（3）多任务

多任务是现代计算机最主要的一个特点，它是指计算机同时执行多个程序，而且各个程序的运行相互独立。Linux 操作系统调试每一个进程平等地访问 CPU。由于 CPU 的处理速度非常快，其结果是启动的应用程序看起来好像是在并行运行。事实上，从 CPU 执行的一个应用程序中的一组指令到 Linux 调试 CPU，与再次运行这个程序之间只有很短的时间延迟，用户是感觉不出来的。

（4）友好的用户界面

Linux 向用户提供了两种界面：用户界面和系统调用界面。Linux 的传统用户界面基于文本的命令行界面，即 Shell。它既可以联机使用，又可以存储在文件上脱机使用。Shell 有很强的程序设计能力，用户可方便地用它编写程序，从而为用户扩充系统功能提供了更高级的手段。Linux 还提供了图形用户界面，它利用鼠标、菜单和窗口等设施，给用户呈现一个直观、易操作、交互性强的友好图形化界面。

（5）设备独立性

设备独立性是指操作系统把所有外部设备统一当做文件来看，只要安装它们的驱动程序，任何用户都可以像使用文件那样操作并使用这些设备，而不必知道它们的具体存在形式。设备独立性的关键在于内核的适应能力，其他的操作系统只允许一定数量或一定种类的外部设备连接，因为每一个设备都是通过其与内核的专用连接独立地进行访问的。Linux 是设备独立的操作系统，它的内核具有高度的适应能力，随着更多程序员加入 Linux 编程，会有更多硬件设备加入到各种 Linux 内核和发行版本中。

（6） 丰富的网络功能

完善的内置网络是 Linux 的一大特点，Linux 在通信和网络功能方面优于其他操作系统。其他操作系统不包含如此紧密的内核结合在一起的连接网络的能力，也没有内置这些联网特性的灵活性。而 Linux 为用户提供了完善的、强大的网络功能。

1） 支持 Internet：Linux 免费提供了大量支持 Internet 的软件，Internet 是在 UNIX 领域中建立并发展起来的，在这方面使用 Linux 是相当方便的，用户能用 Linux 与世界上其他人通过 Internet 网络进行通信。

2） 文件传输：用户能通过一些 Linux 命令完成内部信息或文件的传输。

3） 远程访问：Linux 为系统管理员和技术人员提供了访问其他系统的窗口。通过这种远程访问的功能，一位技术人员能够有效地为多个系统服务，即使那些系统位于很远的地方。

（7） 可靠的安全性

Linux 操作系统采取了许多安全措施，包括对读、写操作进行权限控制，带保护的子系统，审计跟踪和内核授权，这为用户提供了必要的安全保障。

（8） 良好的可移植性

可移植性是指将操作系统从一个平台转移到另一个平台，使它仍然能按其自身的方式运行的能力。Linux 是一款具有良好可移植性的操作系统，能够在微型计算机到大型计算机的任何环境中和平台上运行。该特性为 Linux 操作系统的不同计算机平台与其他任何机器进行准确而有效地通信提供了保障，不需要另外增加特殊的通信接口。

（9） X Window 系统

X Window 系统是用于 UNIX 机器的一个图形系统，该系统拥有强大的界面系统，并支持许多应用程序，是业界标准界面。

（10） 内存保护模式

Linux 使用处理器的内存保护模式来避免进程访问分配给系统内核或者其他进程的内存。对于系统安全来说，这是一个主要的贡献，一个不正确的程序因此不能够再使用系统而崩溃（在理论上）。

（11） 共享程序库

共享程序库是一个程序工作所需要的例程的集合，有许多同时被多于一个进程使用的标准库，因此使用户觉得需要将这些库的程序载入内存一次，而不是一个进程一次，通过共享程序库使这些成为可能，因为这些程序库只有当进程运行的时候才被载入，所以它们被称为动态链接库。

1.1.2 Linux 版本说明

Linux 的版本可以分为两类：内核（Kernel）版本与发行版本（Distribution）版本。内核版本是指在 Linux 的领导下，开发小组开发出来的系统内核版本号。而一些组织或公司将 Linux 内核与应用软件和文档包装起来，并提供一些安装界面、系统设置与管理工具，这样就构成了一个发行版本。常见的发行版本有 Red Hat Linux、Mandriva Linux、Debian Linux 和国产的红旗 Linux 等。

1. Red Hat Linux

Read Hat 最早由 Bob Young 和 Marc Ewing 在 1995 年创建，目前 Red Hat 分为两个系列：由 Red Hat 公司提供收费技术支持和更新的 Red Hat Enterprise Linux，以及由社区开发的免费的 Fedora Core。

Red Hat Linux 是一个比较成熟的 Linux 版本，无论是在销售上还是在装机量上都比较成功。该版本从 4.0 时就开始同时支持 Intel、Alpha 和 Sparc 硬件平台，并且通过 Red Hat 公司的开发，使得用户可以轻松地进行软件升级并彻底卸载应用软件和系统部件。Red Hat Enterprise Linux 是一个收费的操作系统，它适用于服务器；而 Fedora Core 是一个免费版本，该版本提供了最新的软件包，并且其版本的更新周期也非常短，只有 6 个月，目前最新版本为 Fedora Core 6，本书将以该版本为基础全面讲解 Linux 操作系统的相关知识。

2. Mandriva Linux

国内最早开始流行 Linux 操作系统时，Mandriva 就非常流行。最早的 Mandriva 原名为 Mandrake，其开发者是基于 Red Hat 进行开发的。Red Hat 采用 GNOME 桌面系统，而 Mandrake 采用了 KDE。由于安装时 Linux 比较复杂，不适合第一次接触 Linux 的新手，所以 Mandrake 简化了系统安装过程。不但如此，该版本当时还在易用性方面下了不少工夫，包括默认情况下的硬件检测等，这也是当时能在国内流行的原因之一。

3. Debian Linux

Debian 最早由 Ian Murdock 于 1993 年创建，可以称得上是迄今为止最遵循 GNU 规范的 Linux 操作系统。该版本有 3 个系统分支：Stable、Testing 和 Unstable。到 2005 年 5 月，3 个版本分别为 Woody、Sarge 和 Sid。其中，Unstable 为最新测试版本，其中包括最新的软件包，但是也有相对较多的 Bug，适合桌面用户；而 Testing 版本经过 Unstable 中的测试，相对较为稳定，也支持了不少新技术；Woody 一般只用于服务器，上面的版本大部分都比较过时，但是稳定性和安全性都非常高。

4. 红旗 Linux

红旗 Linux 中文操作系统是由中国科学软件所、北大方正电子有限公司和康柏计算机公司联合推出的具有自主版权的全中文化 Linux 发行版本。

红旗 Linux 以全新优化整合的 KDE 图形环境、桌面设计、结构布局和完整和谐的菜单设计，令人耳目一新；集成的硬件自动检测功能，满足 PC 用户硬件的随时更换；高质量的中文字体显示以及高效率的文字输入法选择，确保用户系统办公的工作品质；高效完善的网络使用功能；快捷友好的打印机管理和配置工具；人性化设计的在线升级工具、身份注册、软件更新、数据库管理一线完成，用户可实时提升系统性能、定制个性化桌面环境、拥有完善的工作平台；图形图像软件从基本的 PS/PDF 文件阅读工具到看图、画图、截图，再到图像的扫描、数码相机支持，全线集成满足了用户的各种需求。

1.1.3　各种安装方法介绍

Linux 操作系统的安装方式可以分为 3 种：从硬盘安装、从光盘安装和网络安装。用户可以根据自身实际情况来选择合适的安装方式。例如所使用的计算机是一台工作站，那么用户可以选择以网络方式进行安装；如果是个人计算机，用户可以选择从光盘安装或从硬盘安装。

1. 硬盘安装

在 Windows 操作系统下，将前面下载到的 4 张安装镜像文件复制到硬盘中，再启动计算机进入 DOS 方式，然后将当前工作目录切换到 Linux 安装程序所在的目录中，根据 Linux 安装程序的提示进行逐步安装。从硬盘安装的方式，需要 Linux 与 Windows 共存，比较其他的安装方式有特别的设置方法。

2. 网络安装

从网络安装的方式可以选择从 FTP 或 HTTP 站点安装，不过就个人计算机而言，一般是选择从硬盘安装或光盘安装，这里主要介绍从光盘安装的方式。

3. 光盘安装

从光盘安装操作系统，是个人计算机中的常用方式，本书在介绍安装 Linux 操作系统时，采用在虚拟机中从光盘安装的方式。首先启动计算机，同时按 Del 键，进入到 BIOS 下，设置第一启动驱动器为 CD-ROM，对于该步骤不同的主板应该选择位于不同的位置，用户可参照主板说明书进行设置。

设置完毕，保存并退出 BIOS，计算机重新启动，Linux 的安装光盘可以自动引导计算机进入 Linux 的安装界面。下面的内容中会有详细的介绍。

1.2　企业项目场景 1：Linux 系统安装

宁波四海进出口公司准备安装两台服务器作为企业的文件服务器、数据库服务器、邮件服务器和 Web 服务器。要求系统能稳定可靠地运行，安装维护费用低廉。

考虑问题如下：

1）选用什么操作系统（Enterprise Red Hat Linux 4.0）？

2）安装哪些软件包？

3）如何分配服务器负载？

4）如何安装 Linux 系统 * ？

5）服务器硬件选择（可选）？

任务验收标准：

1）两台服务器分别安装好作为文件服务、数据库服务、邮件服务和 Web 服务软件包。

2）留出足够的用户空间，作为用户使用可系统开发。

3）便于系统重装恢复又能保存用户数据的分区设置。

4）快捷的安装服务。

1.3　Linux 系统安装教学目标（4 学时）

1. 教学目标

最终目标：能根据需求安装 Linux 操作系统。

促成目标：

1）知道 Linux 操作系统的各种版本；

2）能根据用户要求划分硬盘空间；

3）熟悉其他操作系统并存的处理方法；
4）熟悉从各种渠道获取Linux系统安装介质。

2. 工作任务

1）获取和制作Linux操作系统的安装介质（可选）；
2）确定安装方案（单操作系统还是多操作系统）；
3）确定硬盘空间分配和备份方案；
4）Linux系统安装；
5）安装完成后的验证测试。

3. 活动设计

设计一份Linux系统安装和硬盘分配、备份方案。
根据设计方案进行Linux系统安装。

4. 相关理论知识

磁盘阵列技术——硬盘镜像（RAID 1）。

5. 相关实践知识

虚拟机系统安装。

1.4　项目一　实验指导

1.4.1　新建虚拟机

选择“新建虚拟机”→“典型”→“Linux”；
选择Linux版本为“Red Hat Enterprise Linux 4”；
选择虚拟机名称及安装目录，如名称为rhel4，目录自选；
选择网络连接为“使用桥接网络”；
选择磁盘大小容量为“5G”以上；
选择完成。

1.4.2　在虚拟机上安装Linux系统

双击设备中的“CD-ROM”，选择“使用ISO镜像”，并指定ISO安装文件（Red Hat Enterprise 4 安装盘镜像文件1）。

单击命令中的“启动虚拟机”，开始安装。

1）在boot：提示符下按“回车”，开始安装；
2）在“CD Found”对话框选择“Skip”，跳过磁盘检测；
3）在“Welcome欢迎界面”，选择“Next”继续；
4）在“Language Selection”界面选择“Chinese（Simplified）（简体中文）”；
5）在键盘配置选择“默认”，按“下一步”；
6）在磁盘分区设置选择“手工分区”，并确定删除所有数据；
7）在磁盘设置中新建“交换区”和“根”；
选择“文件系统类型”为Swap，大小为500M；

选择“挂接点”为“/”，大小为“使用全部可用空间”；

8）在引导装载程序配置选择“默认”，按“下一步”；

9）在网络配置选择“默认”，按“下一步”；

10）在防火墙配置选择“默认”，按“下一步”；

11）在附加语言支持选择“默认”，按“下一步”；

12）在时区选择“亚洲/上海”；

13）在设置根口令设置口令为“123456”（统一）；

14）在软件包默认安装中选择“定制”；

15）在选择软件包窗口选择所需组件；

桌面：

X 窗口系统

GNOME 桌面环境

应用程序：

编辑器

服务器：

服务器配置工具

万维网服务器

Windows 文件服务器

FTP 服务器

遗留网络服务器

开发：

开发工具

X 软件开发

GNOME 软件开发

KDE 软件开发

系统：

管理工具

系统工具

16）即将安装，按“下一步”开始（不可逆）；

17）在“需要安装的介质”对话框按“继续”；

18）在安装中需要换盘，按“摘要视图（倒数第 3 个）”，双击“CD-ROM”，选择所需 ISO 镜像文件，按“控制视图（最后 1 个）”返回；

19）在安装完成界面，按重新引导；

20）重启后，在欢迎界面，按“下一步”；

21）在许可协议，选“同意”，按“下一步”；

22）在日期和时间页面，选择正确的日期和时间；

23）在显示页面，选择“默认”，按“下一步”；

24）在系统用户，按“下一步”，继续（不创建用户）；

25）在声卡页面，可播放测试声音，按“下一步”；

26）在额外光盘页面，按“下一步”（不安装其他光盘）；
27）在结束设置页面，按“下一步”，设置完毕，准备进入系统；
28）在登录界面，用户名“root”，口令“123456”登录。

1.5　项目一完成步骤详解

1.5.1　新建虚拟机

打开虚拟机应用程序，选择“新建虚拟机”→“典型”→“Linux”。欢迎虚拟机安装界面如图 1-1 所示。

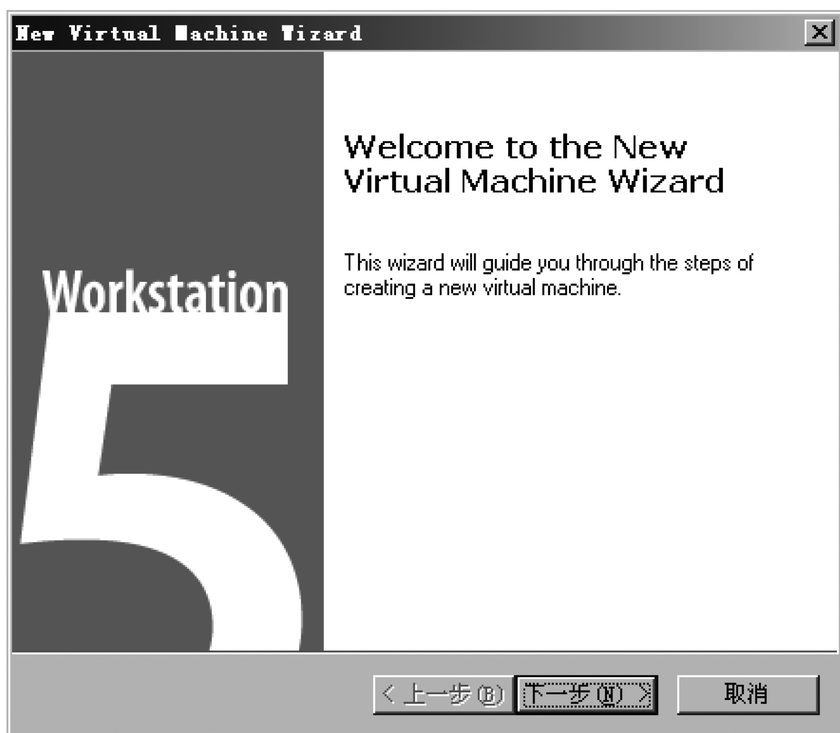


图 1-1　欢迎虚拟机安装界面

选择 Linux 版本为“Red Hat Enterprise Linux 4”。选择操作系统类型如图 1-2 所示。

选择虚拟机名称及安装目录，如名称为 rhel4，目录自选。选择安装路径如图 1-3 所示。

选择网络连接为“使用桥接网络”。选择网络连接方式如图 1-4 所示。

选择磁盘大小容量为“5G”以上。选择安装空间大小如图 1-5 所示。

选择完成后，出现新建好的虚拟机。虚拟安装完成情况如图 1-6 所示。

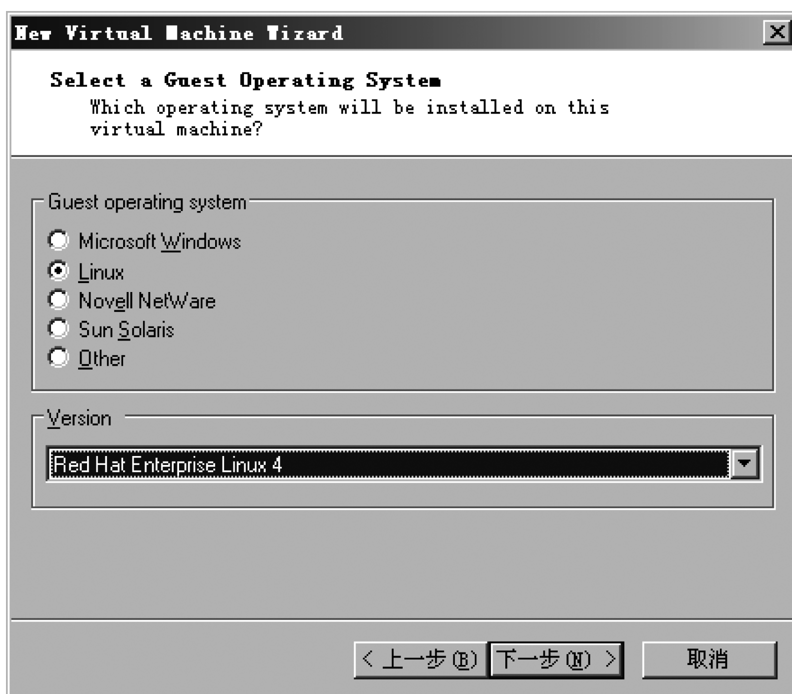


图 1-2 选择操作系统类型

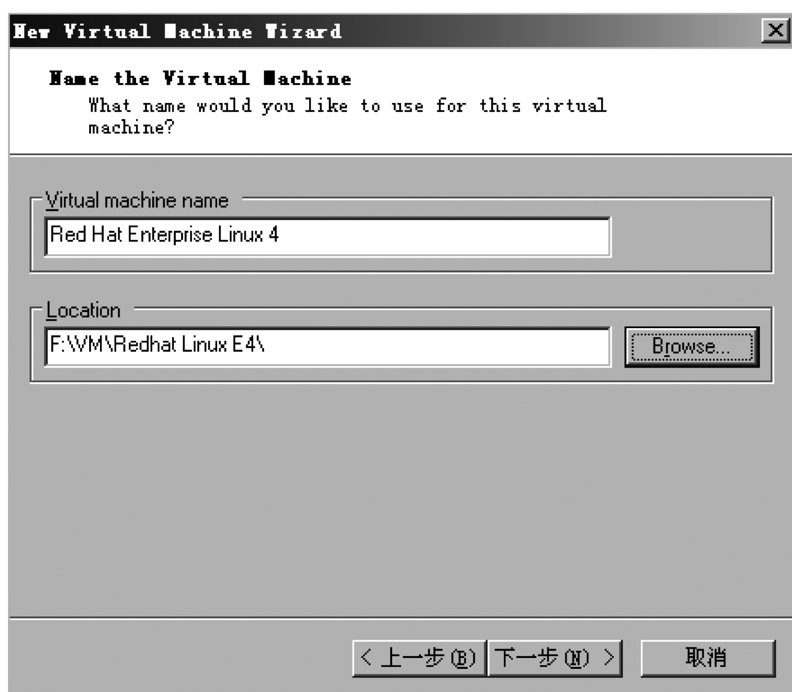


图 1-3 选择安装路径

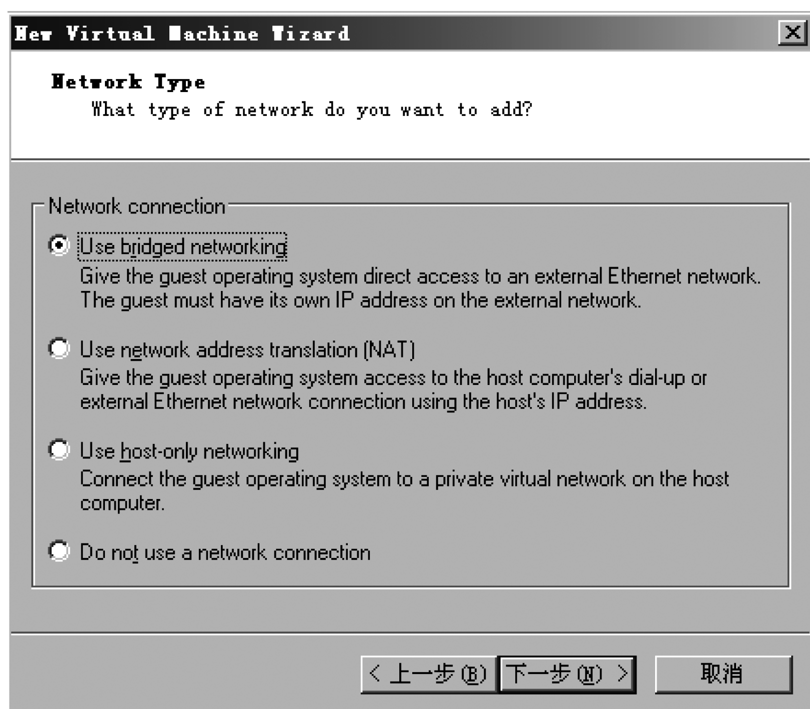


图 1-4　选择网络连接方式

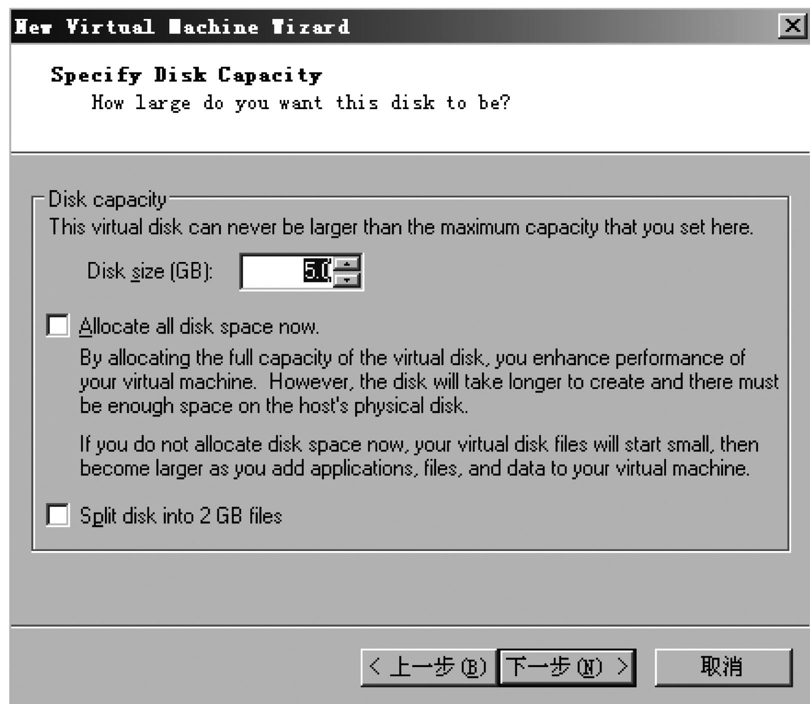


图 1-5　选择安装空间大小

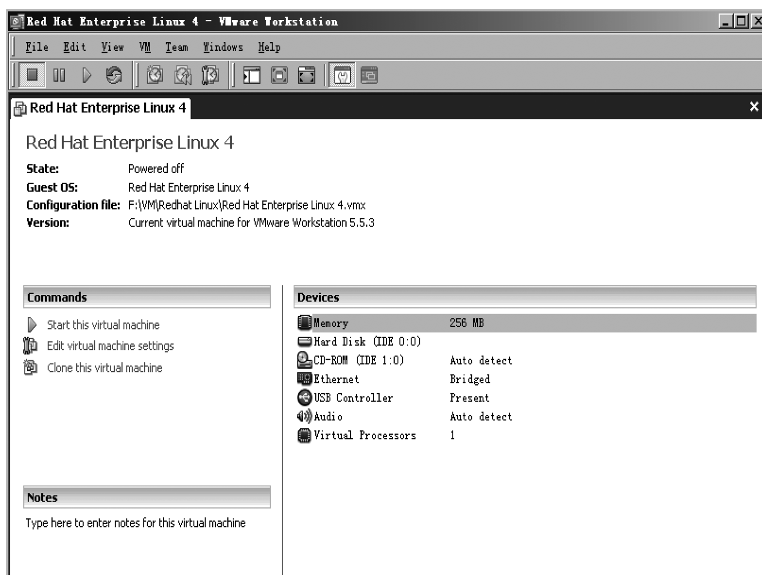


图 1-6　虚拟安装完成情况

1.5.2　在虚拟机上安装 Linux 系统

双击设备中的“CD-ROM”，选择“使用 ISO 镜像”，并指定 ISO 安装文件（Red Hat Enterprise Linux 4 安装盘镜像文件 1）。

单击命令中的“启动虚拟机”，开始安装。

1）在 boot：提示符下按回车，开始安装；

2）在“CD Found”对话框选择“Skip”，跳过磁盘检测（见图 1-7）；

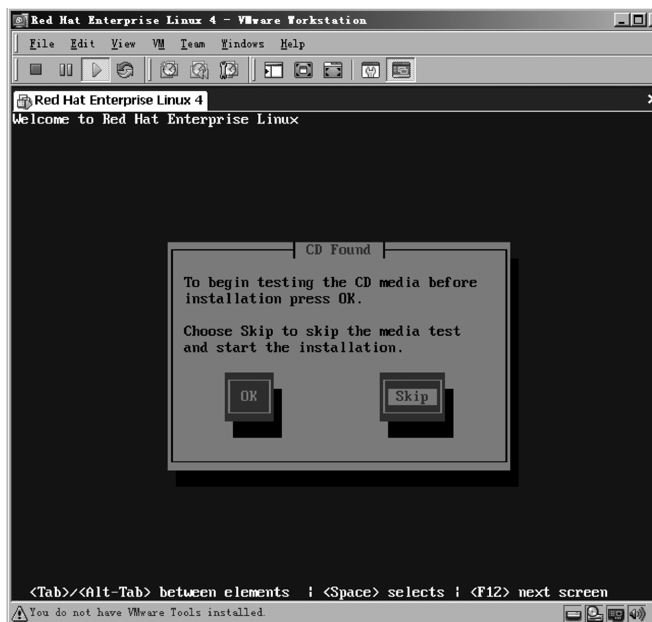


图 1-7　选择是否检测磁盘

3）在“Welcome 欢迎界面”选择“Next”继续（见图 1-8）；



图 1-8　欢迎安装 Linux 界面

4）在“Language Selection”界面选择“Chinese（Simplified）（简体中文）”（见图 1-9）；

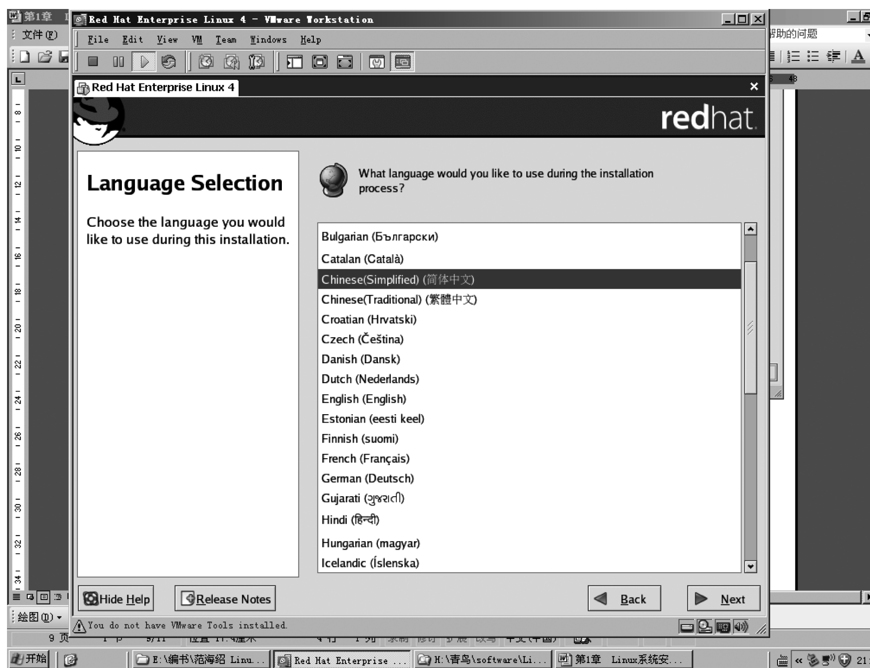


图 1-9　选择语言

5）在键盘配置选择默认，按“下一步”；

6）在磁盘分区设置选择“手工分区”，并确定删除所有数据（见图 1-10）；



图 1-10 选择自动或手动进行磁盘分区

7）在磁盘设置中新建“交换区” 和“根”；

选择“文件系统类型”为“swap”，大小为 500M（见图 1-11）；



图 1-11 交换区设置

选择“挂载点”为“/”，大小为2000M（见图1-12）；



图1-12 根分区设置

选择“挂载点”为“/usr”，用来存放用户数据，大小为“使用全部可用空间”（见图1-13）；



图1-13 用户分区设置

8）在引导装载程序配置选择默认，按“下一步”；
9）在网络配置选择默认，按“下一步”；
10）在防火墙配置选择默认，按“下一步”；
11）在附加语言支持选择默认，按“下一步”；
12）在时区选择“亚洲/上海”；
13）在设置根口令设置口令为“123456”（统一）（见图1-14）；

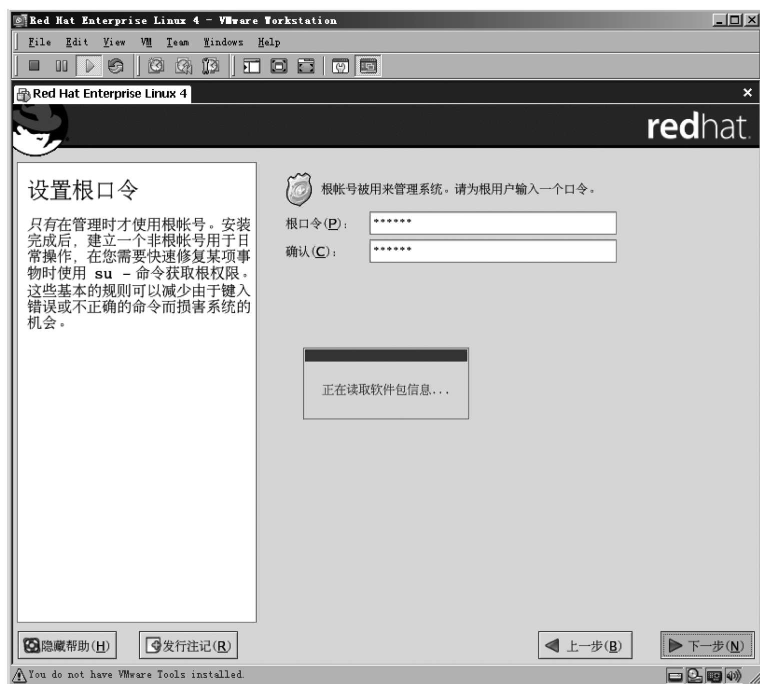


图 1-14　系统口令设置

14）在软件包默认安装中选择“定制”（见图 1-15）；



图 1-15　软件包安装方式选择

15）在选择软件包窗口选择所需组件（见图 1-16）；

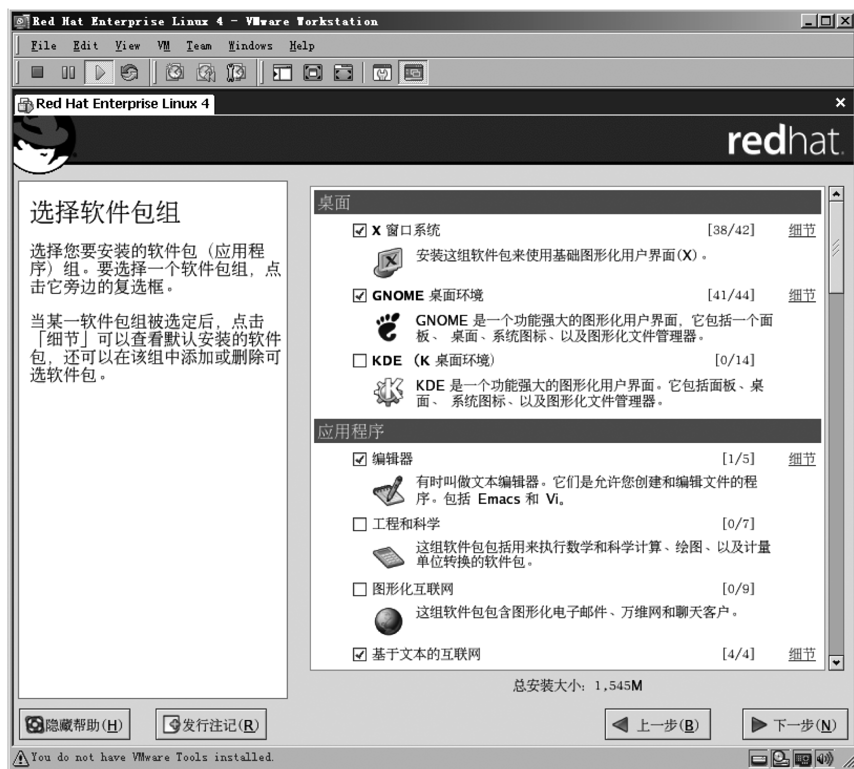


图 1-16　选择软件包

桌面：

X 窗口系统

GNOME 桌面环境

应用程序：

编辑器

服务器：

服务器配置工具

万维网服务器

Windows 文件服务器

FTP 服务器

遗留网络服务器

开发：

开发工具

X 软件开发

GNOME 软件开发

KDE 软件开发

系统：

管理工具

系统工具

16）即将安装，按“下一步”开始（不可逆）；

17）在“需要的安装介质”对话框按“继续”（见图 1-17）；



图 1-17　安装确认

18）在安装中需要换盘，按“摘要视图（倒数第 3 个）”，双击“CD-ROM”，选择所需 ISO 镜像文件，按“控制视图（最后 1 个）”返回；

19）在安装完成界面，按重新引导（见图 1-18）；



图 1-18　安装完成

20）重启后，在欢迎界面，按“下一步”（见图 1-19）；



图 1-19　启动界面

21）在许可协议，选“同意”，按“下一步”（见图 1-20）；

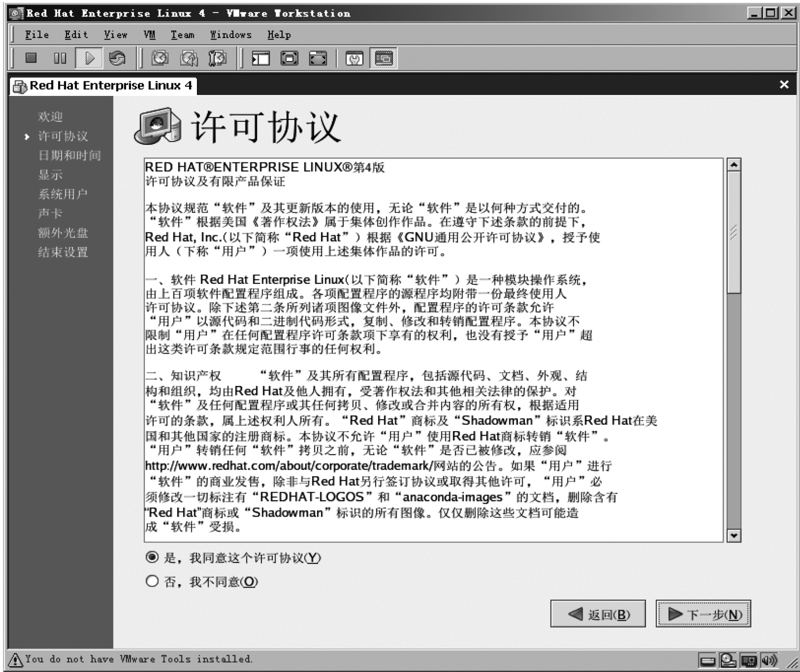


图 1-20　许可协议确认

22）在日期和时间页面，选择正确的日期和时间，按“下一步”；
23）在显示页面，选择默认，按“下一步”；
24）在系统用户，按“下一步”，继续（不创建用户）（见图 1-21）；



图 1-21 额外添加用户界面

25）在声卡页面，可“播放测试声音”，按“下一步”（见图 1-22）；



图 1-22 声卡测试界面

26）在额外光盘页面，按“下一步”（不安装其他光盘）；
27）在结束设置页面，按“下一步”，设置完毕，准备进入系统（见图 1-23）；
28）在登录界面，用户名“root”，口令“123456”登录（见图 1-24、图 1-25）。



图 1-23　初始设置结束界面

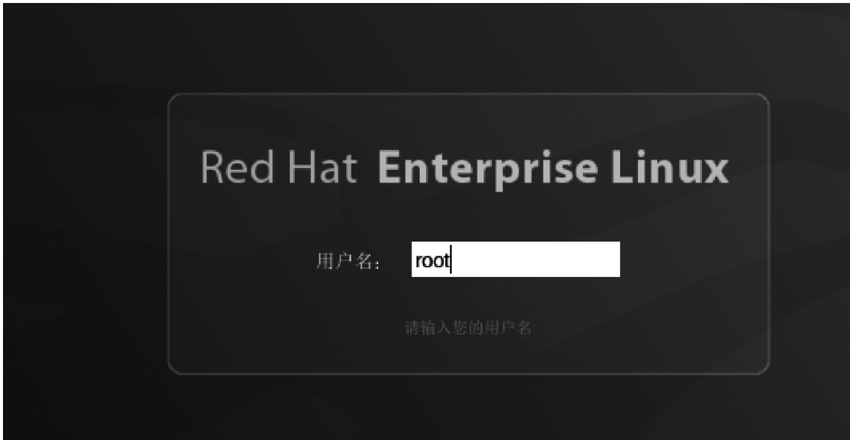


图 1-24　用户登录界面

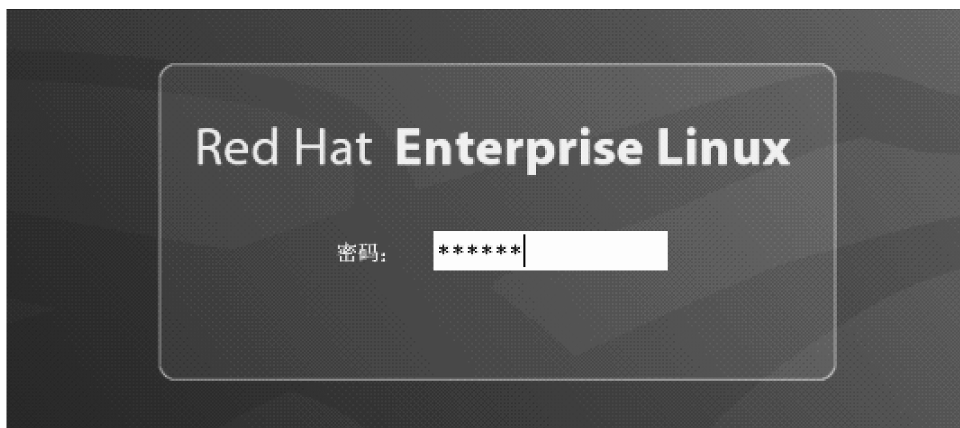


图 1-25　输入密码

1.6　思考题

1）新建虚拟机的网络类型的时候，桥接模式和 NAT（Network Address Translation，网络地址转换）模式的区别在哪里？

2）Linux 的安装类型有哪几种？

第2讲 基本网络配置

 导读

通过对操作系统进行网络配置，从而实现网络的连接和管理。基本的网络配置包括：网卡设置、IP 地址分配和设置、默认路由设置、指定路由设置。

2.1 引言

随着计算机应用的深入，特别是家用计算机越来越普及，一方面希望众多用户能共享信息资源，另一方面也希望各计算机之间能互相传递信息进行通信。个人计算机的硬件和软件配置一般都比较低，其功能也有限，因此要求大型与巨型计算机的硬件和软件资源，以及它们所管理的信息资源应该为众多的微型计算机所共享，以便充分利用这些资源。基于这些原因，促使计算机向网络化发展，将分散的计算机连接成网，组成计算机网络。

2.1.1 网络的定义

计算机网络是现代通信技术与计算机技术相结合的产物。所谓计算机网络，就是把分布在不同地理区域的计算机与专门的外部设备用通信线路互联成一个规模大、功能强的网络系统，从而使众多的计算机可以方便地互相传递信息，共享硬件、软件、数据信息等资源。通俗地说，网络就是通过电缆、电话线或无线通信等互联的计算机的集合。

通过网络，您可以和其他连到网络上的用户一起共享网络资源，如磁盘上的文件及打印机、调制解调器等，也可以和他们互相交换数据信息。

按计算机联网的区域大小，我们可以把网络分为局域网（Local Area Network，LAN）和广域网（Wide Area Network，WAN）。LAN 是指在一个较小地理范围内的各种计算机网络设备互联在一起的通信网络，可以包含一个或多个子网，通常局限在几千米的范围之内。如在一个房间、一座大楼，或是在一个校园内的网络就称为局域网。WAN 连接地理范围较大，常常是一个国家或是一个洲。其目的是为了让分布较远的各局域网互联。我们平常讲的Internet就是最大最典型的广域网。

那么，网络上的计算机之间又是如何交换信息的呢？就像我们说话用某种语言一样，在网络上的各台计算机之间也有一种语言，这就是网络协议，不同的计算机之间必须使用相同的网络协议才能进行通信。当然，网络协议也有很多种，具体选择哪一种协议则要看情况而

定。Internet 上的计算机使用的是 TCP/IP（Transmission Control Protocol/Intermet Protocal，传输控制协议/互联网络协议）。

2.1.2 网络组成

要构成 LAN，必须有其基本部件。LAN 既然是一种计算机网络，自然少不了计算机，特别是个人计算机（PC）。几乎没有一种网络只由大型机或小型机构成。因此，对于 LAN 而言，个人计算机是一种必不可少的构件。计算机互联在一起，当然也不可能没有传输媒体，这种媒体可以是同轴电缆、双绞线、光缆或辐射性媒体。第三个构件是任何一台独立计算机通常都不配备的网卡，也称为网络适配器，但在构成 LAN 时，则是不可少的部件。第四个构件是将计算机与传输媒体相连的各种连接设备，如 RJ- 45 插头座等。具备了上述 4 种网络构件，便可将 LAN 工作的各种设备用媒体互联在一起搭成一个基本的 LAN 硬件平台。

有了 LAN 硬件环境，还需要控制和管理 LAN 正常运行的软件，即 NOS 是在每个 PC 原有操作系统上增加网络所需的功能。例如，当需要在 LAN 上使用字处理程序时，用户的感觉犹如没有组成 LAN 一样，这正是 LAN 操作发挥了对字处理程序访问的管理。在 LAN 情况下，字处理程序的一个复制通常保存在文件服务器中，并由 LAN 上的任何一个用户共享。由上面介绍的情况可知，组成 LAN 需要下述 5 种基本结构：

1）计算机（特别是 PC）；
2）传输媒体；
3）网络适配器；
4）网络连接设备；
5）网络操作系统。

计算机是我们最熟悉不过的了，就不再介绍了，其他部分我们将详细介绍。

2.1.3 网络 IP 地址

IP 地址标识着网络中一个系统的位置。我们知道每个 IP 地址都是由两部分组成的：网络号和主机号。其中，网络号标识一个物理的网络，同一个网络上所有主机需要同一个网络号，该号在互联网中是唯一的；而主机号确定网络中的一个工作端、服务器、路由器其他 TCP/IP 主机。对于同一个网络号，主机号是唯一的。每个 TCP/IP 主机由一个逻辑 IP 地址确定。

IP 地址有两种表示形式：二进制表示（1 和 0 太多了就搞不清）和点分十进制表示。每个 IP 地址的长度为 4 字节，由 4 个 8 位域组成，我们通常称之为八位体。八位体由句点分开，表示为一个 0 ~ 255 之间的十进制数。一个 IP 地址的 4 个域分别标明了网络号和主机号。

TCP/IP 上的每台主机都需要用一个子网屏蔽号。它是一个 4 字节的地址，用来封装或“屏蔽”IP 地址的一部分，以区分网络号和主机号。当网络还没有划分为子网时，可以使用默认的子网屏蔽；当网络被划分为若干个子网时，就要使用自定义的子网屏蔽。

我们来看看默认的子网屏蔽值，它用于一个还没有划分子网的网络。即使是在一个单段网络上，每台主机也都需要这样的默认值。

它的形式依赖于网络的地址类型。在它的 4 个字节里，所有对应网络号的位都被置为 1，于是每个八位体的十进制值都是 255；所有对就主机号的位都置为 0。例如：C 类网地址 192. 168. 0. 1 和相应的默认屏蔽值 255. 255. 255. 0。

我们把屏蔽值和 IP 地址值做“与”的操作其实是一个内部过程，它用来确定一个数据包是传给本地还是远程网络上的主机。其相应的操作过程是这样的：当 TCP/IP 初始化时，主机的 IP 地址和子网屏蔽值相“与”。在数据包被发送之前，再把目的地址也和屏蔽值作“与”，这样如果发现源 IP 地址和目的 IP 地址相匹配，IP 就知道数据包属于本地网上的某台主机；否则数据包将被送到路由器上。

注：我们知道“与”操作是将 IP 地址中的每一位与子网屏蔽中相应的位按逻辑与作比较。

2. 1. 4　网络路由

路由就是选择一条数据包传输路径的过程。当 TCP/IP 主机发送 IP 数据包时，便出现了路由，且当到达 IP 路由器还会再次出现。路由器是从一个物理网向另一个物理网发送数据包的装置，路由器通常被称为网关。对于发送的主机和路由器而言，必须决定向哪里转发数据包。在决定路由时，IP 层查询位于内存中的路由表。

1）当一个主机试图与另一个主机通信时，IP 首先决定目的主机是一个本地网还是远程网。

2）如果目的主机是远程网，IP 将查询路由表来为远程主机或远程网选择一个路由。

3）若未找到明确的路由，IP 用默认的网关地址将一个数据传送给另一个路由器。

4）在该路由器中，路由表再次为远程主机或网络查询路由，若还未找到路由，该数据包将发送到该路由器的默认网关地址。

每发现一条路由，数据包被转送至下一级路由器，称为一次“跳步”，并最终发送至目的主机。若未发现任何一个路由，源主机将收到一个出错信息。

2. 1. 5　网络域名

域名是在 Internet 中用于解决地址对应问题的一种方法。用 IP 地址也可以访问网站空间，但要通过一串长长的数字组成的 IP 地址来访网站。由于 IP 地址不容易被人记忆，采用域名来代替 IP 地址标识站点地址。域名的出现，让访客更容易找到要访问的网站。举例说明：上海快网网站的域名：kwww. cn，kwww 就是域名频道网站的名字，. cn 是中国的意思（. net 网络服务、. cn 中国等，后缀代表不同意思。）

域名解析就是域名到 IP 地址的转换过程。域名的解析工作由 DNS（Domain Name Server，域名服务器）完成。

我们知道域名是为了方便记忆而专门建立的一套地址转换系统，要访问一台互联网上的服务器，最终还必须通过 IP 地址来实现，域名解析就是将域名重新转换为 IP 地址的过程。一个域名只能对应一个 IP 地址，而多个域名可以同时被解析到一个 IP 地址。域名解析需要由专门的域名解析服务器（DNS）来完成。

解析过程比如一个域名为 www. kwww. cn，实现 HTTP 服务，如果想看到这个网站，要进行解析，首先在域名注册商那里通过专门的 DNS（域名服务器）解析到一个 Web 服务器的一个固定 IP 上：211. 214. 1. ***，然后通过 Web 服务器来接收这个域名，把

www. kwww. cn 这个域名映射到这台服务器上，那么输入 www. kwww. cn 这个域名就可以访问网站内容，即实现了域名解析的全过程。

人们习惯记忆域名，但机器间互相只认 IP 地址，域名与 IP 地址之间是一一对应的，它们之间的转换工作称为域名解析，域名解析需要由专门的域名解析服务器来完成，整个过程是自动进行的。

当应用过程需要将一个主机域名映射为 IP 地址时，就调用域名解析函数，解析函数将待转换的域名放在 DNS 请求中，kwww. cn 以 UDP（User Datagram Protocal，用户数据报协议）报文方式发给本地域名服务器。本地的域名服务器查到域名后，将对应的 IP 地址放在应答报文中返回。同时域名服务器还必须具有连向其他服务器的信息以支持不能解析时的转发。若域名服务器不能回答该请求，kwww. cn 则此域名服务器就暂成为 DNS 中的另一个客户，向根域名服务器发出请求解析，根域名服务器一定能找到下面的所有二级域名的域名服务器，这样以此类推，一直向下解析，直到查询到所请求的域名。

2. 2 企业项目场景 2：基本网络配置

宁波四海进出口公司搬迁到一座新大楼，需要将多台服务器和工作站重新进行网络设置和联网。其中，服务器均为 Linux 操作系统，请将它们设置好网卡、IP 地址、默认路由、指定路由和简单域名服务。

IP 地址和域名设定如下：

www1. net. cn　　192. 168. 1. 1

www2. net. cn　　192. 168. 1. 2

默认网关为　　192. 168. 1. 254

到达 10. 10. 1. 0 网络的网关为 192. 168. 1. 200

验收标准：

1）每台服务器配置了正确的网卡和 IP 地址。

2）配置了正确的默认网关和指定网关。

3）能用命令检测上述配置。

2. 3 Linux 系统基本网络配置教学目标（8 学时）

1. 教学目标

最终目标：能配置基本网络、测试网络和简单应用网络工具。

促成目标：

1）掌握网卡的设置方法；

2）掌握路由、简单域名的设置方法；

3）掌握常用网络测试与控制命令的使用；

4）掌握文件上传与下载、远程登录命令的操作。

2. 工作任务

安装网卡、设置路由、检测网络配置、设置简单域名服务、设置文件上传下载和远程登

录服务。

3. 活动设计

安装网卡，设置 IP 地址和掩码。

设置默认路由和指定路由，查询路由表。

设置简单域名服务并验证。

设置文件上传下载服务。

设置远程登录服务。

4. 相关理论知识

IP 地址分配设计。

5. 相关实践知识

网络基础知识（路由概念）。

2.4　项目二　实验指导

2.4.1　配置网卡

应用程序→系统设置→网络，打开网络设置对话框。

1）在“设备”页选择“新建”。

在“选择设备类型”中选择“以太网连接”，按“前进”，再选择一个网卡型号，再按“前进”。

2）在弹出的“配置网络设备”窗口中选择“静态设置 IP 地址”，并填写“地址”和“子网掩码”，如：

地址：192.168.1.100

子网掩码：255.255.255.0

填写完成，按“前进”，在“创建以太网设备”窗口按“应用”。

3）在网络配置主窗口按“激活”、“是”激活网卡。若没有出错，网卡状态变成“活跃”，说明配置成功。

4）选择“文件”→“保存”，以保存网络设置。

2.4.2　测试网络

在桌面空白处右击，在弹出菜单中选择“打开终端”，在终端窗口中操作。

重启网络（网络设置新增或改变都要重启网络）

在终端窗口中输入命令：service network restart

一般服务都是用“service”命令启动、停止、重启的，后跟命令分别为“start”、“stop”、“restart”。这里的“network”是服务进程的名称。

2.4.3　用“ping”命令测试

输入命令“ping　192.168.1.100”，若反馈类似为

64 bytes from 192.168.1.100：icmp_seq = 0ttl = 64time = 0.005ms，

说明网卡运行正常，网络通畅。可以用该命令测试其他链接地址，如 Windows 系统 IP 地址，但必须是相同网段的地址。若 Windows 系统地址为 192. 168. 1. 200，则

ping 192. 168. 1. 200

若反馈信息为

From 192. 168. 1. 200 icmp_seq = 0 Destination Host Unreachable，

说明网络不通，或对方设备未开。

2. 4. 4 用 ifconfig 命令

在终端输入命令

ifconfig，“#”是超级用户提示符，普通用户提示符是“ $ ”。

这个命令将输出所有网络设备的配置情况，

eth0 为第一块网卡，第二、第三依次为 eth1 、eth2. . .

lo 为本机回路地址，地址为 127. 0. 0. 1。

“#ifconfig eth0”，这个命令仅输出第一块网卡的配置情况。

2. 4. 5 开通和关闭网络设备（网卡）

“#ifconfig eth0 down”，这个命令将使 eth0 不能与其他网络设备通信，可以通过另开一个终端，使用“ping”命令来验证。

“#ifconfig eth0 up”，这个命令刚好与上面的命令相反，它使 eth0 开通网络，同样可以用“ping”命令来验证。由此可知，可以用“ifconfig”命令来控制网络设备的开通和关闭。

2. 4. 6 设置临时 IP 地址

可以用“ifconfig”命令给网卡设置一个临时 IP 地址，用于测试，例如：

“#ifconfig eth0 192. 168. 1. 150/24”；设置临时地址为 192. 168. 1. 150，掩码长度为 24，即 255. 255. 255. 0。设置完成后仍可用“ifconfig”检查验证，但要切记，这是临时 IP 地址，若重启网络或关机重启则该 IP 地址将丢失。

2. 4. 7 设置默认网关

在网络设置对话框的设备页，选中网卡，按“编辑”，在“以太网设备”的“常规”页中，找到“静态设置 IP 地址”，在“默认网关地址”中填写网关地址，例如：192. 168. 1. 254，按“确定”，并保存网络配置。

要测试网关设置，需要重启网络，用“route”命令检查即可。

```
#service network restart        ；重启网络
#route                          ；检查网关设置
```

显示：

```
Destination    Gateway            Genmask     Flags  ....
Default        192. 168. 1. 254   0. 0. 0. 0   UG   0  0  0  eth0
```

可以看到网关已经设置好了。

2.4.8　设置指定网关

在网络设置中选择“编辑”，在以太网设置中选择“路由”，按“添加”，在弹出的对话框中填写路由情况，以下分别说明：

地址：要到达的网络或主机；

子网掩码：要到达网络或主机的掩码；

网关：指定的网关地址。

上述设置需要遵循网络规则，子网掩码为 255. 255. 255. 255，则该目标为主机，否则为一个网络。

网关一定与本机地址是同网段的，否则无效。

验证方法同默认网关。

2.4.9　文件上传和下载

用 vsftpd 服务，可以实现文件传送。

```
#service vsftpd start        ; 启动 ftp 服务
#iptables  -F                ; 关闭防火墙，也可以通过“安全级别”设置
ftp  IP 地址                 ; 在客户端用 ftp 登录到主机
put  文件名                  ; 上传文件
get  文件名                  ; 下载文件
bye                          ; 退出 ftp 服务
```

也可以通过浏览器登录主机：

```
ftp：//IP 地址               ; 在浏览器中输入 ftp 协议和地址
```

2.4.10　远程登录

在服务中选中 krb5-telnet，并重启 xinetd 服务，关闭防火墙。

在客户端登录：

```
telnet  IP 地址
```

2.5　项目二完成步骤详解

2.5.1　网卡设置

根据要求，将网卡 IP 地址设置为 192. 168. 1. 1，掩码为 255. 255. 255. 0。通过菜单“应用程序”→“系统设置”→“网络”，打开网络配置对话框，如图 2-1 所示。

点击“新建”，进入“选择设备类型”对话框。选择设备类型如图 2-2 所示。

在“设备类型”中选择“以太网连接”，点击“前进”，进入“选择以太网设备”对话框，如图 2-3 所示。

选择合适的以太网卡，点击“前进”，进入“配置网络设置”对话框，选择“静态设置的 IP 地址”，并填写 IP 地址、子网掩码和默认网关，如图 2-4 所示。



图 2-1　网络配置对话框



图 2-2　选择设备类型



图 2-3　选择网卡



图 2-4　设置 IP 地址

点击“前进”，显示“创建以太网设备”对话框。网卡设置完成信息如图 2-5 所示。



图 2-5 网卡设置完成信息

点击“应用”，返回“网络配置”对话框，点击“激活”并确认“是”，激活已配置的网卡，如图 2-6 所示。



图 2-6 激活网卡

在“文件”菜单中选择“保存”。保存设置信息如图 2-7 所示。



图 2-7　保存设置信息

2. 5. 2　设置指定网关

在“网络配置”对话框中，点击“编辑”工具按钮，进入“以太网设备”对话框，然后选择“路由（R）”页面，如图 2-8 所示。



图 2-8　路由设置页面

按“添加（A)”菜单，弹出“添加/编辑 IP 地址”对话框，输入到达网络 10. 10. 1. 0 和指定路由地址 192. 168. 1. 200，如图 2-9 所示。

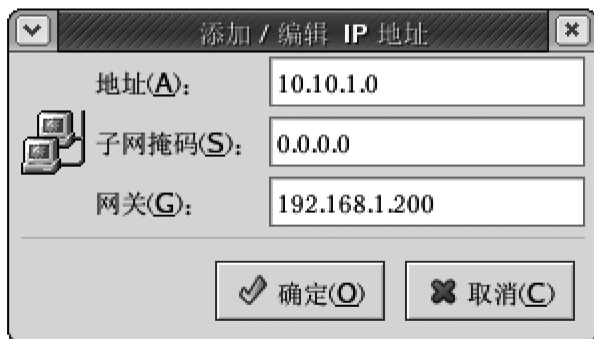


图 2-9　添加路由设置

按“确定”，显示“静态网络路由”。路由添加完成信息如图 2-10 所示。



图 2-10　路由添加完成信息

按“确定”，返回“网络配置”对话框，再次选择“文件”→“保存”，保存所设置的内容。

2. 5. 3　简单域名设置

在“网络配置对话框”，选择“主机”页面，并点击“新建”工具按钮，弹出“添加/

编辑主机项目”对话框，如图 2-11 所示。



图 2-11　添加简单域名解析

分别输入两个 IP 地址、主机名和别名，按“确定”，显示所配置的域名，如图 2-12 所示。



图 2-12　简单域名解析设置

也可以通过编辑文件/etc/hosts，实现同样的功能。用 vi 编辑的方法是，在终端窗口输入命令，如图 2-13 所示。



图 2-13　编辑简单域名解析文件

然后编辑该文件，如图 2-14 所示。

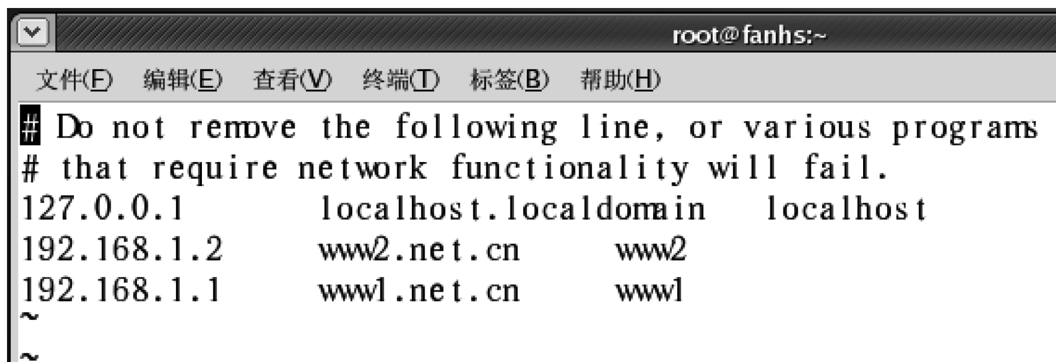


图 2-14　编辑域名解析文件内容

编辑后保存文件。

2.5.4　网络测试

1. 网络配置检查

在终端窗口执行 ifconfig　eth0 命令，显示网卡的配置情况，如图 2-15 所示。



图 2-15　执行 ifconfig　eth0 命令

用 ping 命令检查网卡的导通情况，如图 2-16 所示。

```
root@fanhs:~
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@fanhs ~]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=0 ttl=64 time=0.125 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.070 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.070 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.065 ms
64 bytes from 192.168.1.1: icmp_seq=4 ttl=64 time=1.09 ms

--- 192.168.1.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 399
rtt min/avg/max/mdev = 0.065/0.285/1.099/0.407 ms, pipe 2
[root@fanhs ~]#
```

图 2-16 执行 ping 命令

2. 测试简单域名配置

用 ping 命令测试所配置的域名，如图 2-17 所示。

```
[root@fanhs ~]# ping www1.net.cn
PING www1.net.cn (192.168.1.1) 56(84) bytes of data.
64 bytes from www1.net.cn (192.168.1.1): icmp_seq=0 ttl=64 time=0.1
64 bytes from www1.net.cn (192.168.1.1): icmp_seq=1 ttl=64 time=0.(
64 bytes from www1.net.cn (192.168.1.1): icmp_seq=2 ttl=64 time=1.1

--- www1.net.cn ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2001ms
rtt min/avg/max/mdev = 0.072/0.449/1.148/0.494 ms, pipe 2
[root@fanhs ~]# ping www1
PING www1.net.cn (192.168.1.1) 56(84) bytes of data.
64 bytes from www1.net.cn (192.168.1.1): icmp_seq=0 ttl=64 time=0.(
64 bytes from www1.net.cn (192.168.1.1): icmp_seq=1 ttl=64 time=1.(
```

图 2-17 执行 ping 简单域名命令

3. 测试路由配置

首先用 service network restart 命令重启网络，然后用 route 命令检查路由配置，如图 2-18 所示。

```
root@fanhs:~
文件(F) 编辑(E) 查看(V) 终端(T) 标签(B) 帮助(H)
[root@fanhs ~]# service network restart
正在关闭接口 eth0:                                    [  确定  ]
关闭环回接口:                                         [  确定  ]
设置网络参数:                                         [  确定  ]
弹出环回接口:                                         [  确定  ]
弹出界面 eth0:                                        [  确定  ]
[root@fanhs ~]# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use
192.168.1.0     *               255.255.255.0   U     0      0        0
10.10.1.0       192.168.1.200   255.255.255.0   UG    0      0        0
169.254.0.0     *               255.255.0.0     U     0      0        0
default         192.168.1.254   0.0.0.0         UG    0      0        0
[root@fanhs ~]#
```

图 2-18 重启网络命令

2.6 思考题

1）Ifconfig 命令的功能有哪些？如何使用？

2）Linux 简单域名服务如何实现，配置文件和格式是怎样的？

3）默认路由和指定路由是怎样设置的，如何验证？

第3讲 Linux 系统用户管理

3.1 引言

Linux 是一个多任务、多用户的操作系统，要能做到不同的用户能同时访问不同的文件，允许不同的用户从本地登录或远程登录，这时用户必须拥有一个合法的账号，Linux 系统正是通过账号来实现对用户的访问进行控制的，因此，需要对用户与组进行有效的管理。Linux 下的用户可以分为 3 类：超级用户、系统用户和普通用户。超级用户的用户名为 root，它具有一切权限，只有进行系统维护（例如建立用户等）或在其他必要情形下才用超级用户登录，以避免系统出现安全问题。系统用户是 Linux 系统正常工作所必需的内建的用户，主要是为了满足相应的系统进程对文件属主的要求而建立的，系统用户不能用来登录，如 bin、daemon、adm、lp 等用户。而普通用户是为了让使用者能够使用 Linux 系统资源而建立的，我们的大多数用户属于此类。每个用户都有一个数值，称为 UID。超级用户的 UID 为 0，系统用户的 UID 一般为 1 ~ 499，普通用户的 UID 是 500 ~ 60000 之间的值。

实现用户账号的管理，要完成的工作主要有如下几个方面：

- 用户账号的添加、删除与修改；
- 用户口令的管理；
- 用户组的管理。

1. Linux 系统用户账号的管理

用户账号的管理工作主要涉及用户账号的添加、修改和删除。添加用户账号就是在系统中创建一个新账号，然后为新账号分配用户号、用户组、主目录和登录 Shell 等资源。刚添加的账号是被锁定的，无法使用。

（1）添加新的用户账号使用 useradd 命令

其语法如下：

useradd 选项 用户名

其中各选项含义如下：

-c comment 指定一段注释性描述。

-d 目录 指定用户主目录，如果此目录不存在，则同时使用-m 选项，可以创建主目录。

-g 用户组 指定用户所属的用户组。

-G 用户组，用户组 指定用户所属的附加组。

-s Shell 文件 指定用户的登录 Shell。

-u 用户号 指定用户的用户号，如果同时有-o 选项，则可以重复使用其他用户的标识号。

用户名 指定新账号的登录名。

例 1：# useradd -d /usr/sam -m sam

此命令创建了一个用户 sam。

其中-d 和-m 选项用来为登录名 sam 产生一个主目录/usr/sam（/usr 为默认的用户主目录所在的父目录）。

例 2：# useradd -s /bin/sh -g group -G adm，root gem

此命令新建了一个用户 gem，该用户的登录 Shell 是/bin/sh，它属于 group 用户组，同时又属于 adm 和 root 用户组，其中 group 用户组是其主组。

这里可能新建组：#groupadd group 及 groupadd adm。

增加用户账号就是在/etc/passwd 文件中为新用户增加一条记录，同时更新其他系统文件，如/etc/shadow，/etc/group 等。

Linux 提供了集成的系统管理工具 userconf，它可以用来对用户账号进行统一管理。

（2）删除账号

如果一个用户的账号不再使用，可以从系统中删除。删除用户账号就是要将/etc/passwd 等系统文件中的该用户记录删除，必要时还要删除用户的主目录。删除一个已有的用户账号使用 userdel 命令，其格式如下：

userdel 选项 用户名

常用的选项是-r，它的作用是把用户的主目录一起删除。

例如：# userdel -r sam

此命令删除用户 sam 在系统文件中（主要是/etc/passwd，/etc/shadow，/etc/group 等）的记录，同时删除用户的主目录。

（3）修改账号

修改用户账号就是根据实际情况更改用户的有关属性，如用户号、主目录、用户组、登录 Shell 等。

修改已有用户的信息使用 usermod 命令，其格式如下：

usermod 选项 用户名

常用的选项包括-c，-d，-m，-g，-G，-s，-u 以及-o 等，这些选项的意义与 useradd 命令中的选项一样，可以为用户指定新的资源值。另外，有些系统可以使用如下选项：

-l 新用户名

这个选项指定一个新的账号，即将原来的用户名改为新的用户名。

例如：# usermod -s /bin/ksh -d /home/z -g developer sam

此命令将用户 sam 的登录 Shell 修改为 ksh，主目录改为/home/z，用户组改为 developer。

（4）用户口令的管理

用户管理的一项重要内容是用户口令的管理。用户账号刚创建时没有口令，但是被系统锁定，无法使用，必须为其指定口令后才可以使用，即使是指定空口令。

指定和修改用户口令的 Shell 命令是 passwd。超级用户可以为自己和其他用户指定口令，普通用户只能用它修改自己的口令。命令的格式为

passwd 选项 用户名

可使用的选项：

-l 锁定口令，即禁用账号。

-u 口令解锁。

-d 使账号无口令。

-f 强迫用户下次登录时修改口令。

如果默认用户名，则修改当前用户的口令。

例如，假设当前用户是 sam，则下面的命令修改该用户自己的口令：

$ passwd

old password：******

New password：*******

Re-enter new password：*******

如果是超级用户，可以用下列形式指定任何用户的口令：

passwd sam

New password：*******

Re-enter new password：*******

普通用户修改自己的口令时，passwd 命令会先询问原口令，验证后再要求用户输入两遍新口令，如果两次输入的口令一致，则将这个口令指定给用户；而超级用户为用户指定口令时，就不需要知道原口令。

为了系统安全起见，用户应该选择比较复杂的口令，例如最好使用 8 位长的口令，口令中包含有大写、小写字母和数字，并且应该与姓名、生日等不相同。

为用户指定空口令时，执行下列形式的命令：

passwd -d sam

此命令将用户 sam 的口令删除，这样用户 sam 下一次登录时，系统就不再询问口令。

passwd 命令还可以用-l（lock）选项锁定某一用户，使其不能登录，例如：

passwd -l sam

2. Linux 系统用户组的管理

每个用户都有一个用户组，系统可以对一个用户组中的所有用户进行集中管理。不同 Linux 系统对用户组的规定有所不同，如 Linux 下的用户属于与它同名的用户组，这个用户组在创建用户时同时创建。

用户组的管理涉及用户组的添加、删除和修改。组的增加、删除和修改实际上就是对/etc/group 文件的更新。

（1）增加一个新的用户组使用 groupadd 命令

其格式如下：

groupadd 选项 用户组

可以使用的选项有：

-g GID 指定新用户组的组标识号（GID）。

-o 一般与-g 选项同时使用，表示新用户组的 GID 可以与系统已有用户组的 GID 相同。

例 1：# groupadd group1

此命令向系统中增加了一个新组 group1，新组的组标识号是在当前已有的最大组标识号

的基础上加 1。

例 2：#groupadd -g 101 group2

此命令向系统中增加了一个新组 group2，同时指定新组的组标识号是 101。

（2）如果要删除一个已有的用户组，使用 groupdel 命令

其格式如下：

groupdel 用户组

例如：#groupdel group1

此命令从系统中删除组 group1。

（3）修改用户组的属性使用 groupmod 命令

其语法如下：

groupmod 选项 用户组

常用的选项有：

-g GID 为用户组指定新的组标识号。

-o 与-g 选项同时使用，用户组的新 GID 可以与系统已有用户组的 GID 相同。

-n 新用户组 将用户组的名字改为新名字。

例 1：# groupmod -g 102 group2

此命令将组 group2 的组标识号修改为 102。

例 2：# groupmod -g 10000 -n group3 group2

此命令将组 group2 的标识号改为 10000，组名修改为 group3。

（4）如果一个用户同时属于多个用户组，那么用户可以在用户组之间切换，以便具有其他用户组的权限。用户可以在登录后，使用命令 newgrp 切换到其他用户组，这个命令的参数就是目的用户组。

例如：$ newgrp root

这条命令将当前用户切换到 root 用户组，前提条件是 root 用户组确实是该用户的主组或附加组。类似于用户账号的管理，用户组的管理也可以通过集成的系统管理工具来完成。

3. 与用户账号有关的系统文件

完成用户管理的工作有许多种方法，但是每一种方法实际上都是对有关的系统文件进行修改。与用户和用户组相关的信息都存放在一些系统文件中，这些文件包括/etc/passwd，/etc/shadow，/etc/group 等。下面分别介绍这些文件的内容。

1）/etc/passwd 文件是用户管理工作涉及的最重要的一个文件。Linux 系统中的每个用户都在/etc/passwd 文件中有一个对应的记录行，它记录了这个用户的一些基本属性。这个文件对所有用户都是可读的。它的内容类似下面的例子：

```
# cat /etc/passwd
root：x：0：0：Superuser：/：
daemon：x：1：1：System daemons：/etc：
bin：x：2：2：Owner of system commands：/bin：
sys：x：3：3：Owner of system files：/usr/sys：
adm：x：4：4：System accounting：/usr/adm：
uucp：x：5：5：UUCP administrator：/usr/lib/uucp：
```

auth：x：7：21：Authentication administrator：/tcb/files/auth：

cron：x：9：16：Cron daemon：/usr/spool/cron：

listen：x：37：4：Network daemon：/usr/net/nls：

lp：x：71：18：Printer administrator：/usr/spool/lp：

sam：x：200：50：Sam san：/usr/sam：/bin/sh

从上面的例子我们可以看到，/etc/passwd 中一行记录对应着一个用户，每行记录又被冒号分隔为 7 个字段，其格式和具体含义如下：

用户名：口令：用户标识号：组标识号：注释性描述：主目录：登录 Shell

①“用户名”是代表用户账号的字符串。通常长度不超过 8 个字符，并且由大小写字母和/或数字组成。登录名中不能有冒号，因为冒号在这里是分隔符。为了兼容起见，登录名中最好不要包含点字符（.），并且不使用连字符（－）和加号（＋）打头。

②“口令”系统中，存放着加密后的用户口令字。虽然这个字段存放的只是用户口令的加密串，不是明文，但是由于/etc/passwd 文件对所有用户都可读，所以这仍是一个安全隐患。因此，现在许多 Linux 系统（如 SVR4）都使用了 shadow 技术，把真正的加密后的用户口令字存放到/etc/shadow 文件中，而在/etc/passwd 文件的口令字段中只存放一个特殊的字符，例如“x”或者“*”。

③“用户标识号”是一个整数，系统内部用它来标识用户。一般情况下它与用户名是一一对应的。如果几个用户名对应的用户标识号是一样的，系统内部将把它们视为同一个用户，但是它们可以有不同的口令、不同的主目录以及不同的登录 Shell 等。

通常用户标识号的取值范围是 0 ~ 65535。0 是超级用户 root 的标识号，1 ~ 99 由系统保留，作为管理账号，普通用户的标识号从 100 开始。在 Linux 系统中，这个界限是 500。

④“组标识号”字段记录的是用户所属的用户组。它对应着/etc/group 文件中的一条记录。

⑤“注释性描述”字段记录着用户的一些个人情况，例如用户的真实姓名、电话、地址等，这个字段并没有什么实际的用途。在不同的 Linux 系统中，这个字段的格式并没有统一。在许多 Linux 系统中，这个字段存放的是一段任意的注释性描述文字，用做 finger 命令的输出。

⑥“主目录”，也就是用户的起始工作目录，它是用户在登录到系统之后所处的目录。在大多数系统中，各用户的主目录都被组织在同一个特定的目录下，而用户主目录的名称就是该用户的登录名。各用户对自己的主目录有读、写、执行（搜索）权限，其他用户对此目录的访问权限则根据具体情况设置。

⑦ 用户登录后，要启动一个进程，负责将用户的操作传给内核，这个进程是用户登录到系统后运行的命令解释器或某个特定的程序，即 Shell。Shell 是用户与 Linux 系统之间的接口。Linux 的 Shell 有许多种，每种都有不同的特点。常用的有 sh（Bourne Shell），csh（C Shell），ksh（Korn Shell），tcsh（TENEX/TOPS-20 type C Shell），bash（Bourne Again Shell）等。系统管理员可以根据系统情况和用户习惯为用户指定某个 Shell。如果不指定 Shell，那么系统使用 sh 为默认的登录 Shell，即这个字段的值为/bin/sh。

用户的登录 Shell 也可以指定为某个特定的程序（此程序不是一个命令解释器）。利用这一特点，我们可以限制用户只能运行指定的应用程序，在该应用程序运行结束后，用户就

自动退出了系统。有些 Linux 系统要求只有那些在系统中登记了的程序才能出现在这个字段中。

系统中有一类用户称为伪用户（psuedo users），这些用户在/etc/passwd 文件中也占有一条记录，但是不能登录，因为它们的登录 Shell 为空。它们的存在主要是方便系统管理，满足相应的系统进程对文件属主的要求。常见的伪用户如下所示。

伪　用　户	含　　义
bin	拥有可执行的用户命令文件
sys	拥有系统文件
adm	拥有账户文件
uucp	uucp 使用
lp	lp 或 lpd 子系统使用
nobody	NFS 使用

拥有账户文件

除了上面列出的伪用户外，还有许多标准的伪用户，例如 audit、cron、mail、usenet 等，它们也都各自为相关的进程和文件所需要。

由于/etc/passwd 文件是所有用户都可读的，如果用户的密码太简单或规律比较明显，一台普通的计算机就能够很容易地将它破解，因此对安全性要求较高的 Linux 系统都把加密后的口令字分离出来，单独存放在一个文件中，这个文件是/etc/shadow 文件。只有超级用户才拥有该文件读权限，这就保证了用户密码的安全性。

2）/etc/shadow 中的记录行与/etc/passwd 中的一一对应，它由 pwconv 命令根据/etc/passwd 中的数据自动产生。它的文件格式与/etc/passwd 类似，由若干个字段组成，字段之间用“:”隔开。这些字段是：

登录名，加密口令，最后一次修改时间，最小时间间隔，最大时间间隔，警告时间，不活动时间，失效时间，标志。

①“登录名”是与/etc/passwd 文件中的登录名相一致的用户账号。

②“口令”字段存放的是加密后的用户口令字，长度为 13 个字符。如果为空，则对应用户没有口令，登录时不需要口令；如果含有不属于集合｛./0-9A-Za-z｝中的字符，则对应的用户不能登录。

③“最后一次修改时间”表示的是从某个时刻起，到用户最后一次修改口令时的天数。时间起点对不同的系统可能不一样。例如在 SCO Linux 中，这个时间起点是 1970 年 1 月 1 日。

④“最小时间间隔”指的是两次修改口令之间所需的最小天数。

⑤“最大时间间隔”指的是口令保持有效的最大天数。

⑥“警告时间”字段表示的是从系统开始警告用户到用户密码正式失效之间的天数。

⑦“不活动时间”表示的是用户没有登录活动但账号仍能保持有效的最大天数。

⑧“失效时间”字段给出的是一个绝对的天数，如果使用了这个字段，那么就给出相应账号的生存期。期满后，该账号就不再是一个合法的账号，也就不能再用来登录了。

下面是/etc/shadow 的一个例子：

```
# cat /etc/shadow
root:Dnakfw28zf38w:8764:0:168:7:::
daemon:*::0:0::::
bin:*::0:0::::
sys:*::0:0::::
adm:*::0:0::::
uucp:*::0:0::::
nuucp:*::0:0::::
auth:*::0:0::::
cron:*::0:0::::
listen:*::0:0::::
lp:*::0:0::::
sam:EkdiSECLWPdSa:9740:0:0::::
```

3）用户组的所有信息都存放在/etc/group 文件中。

将用户分组是 Linux 系统中对用户进行管理及控制访问权限的一种手段。每个用户都属于某个用户组；一个组中可以有多个用户，一个用户也可以属于不同的组。当一个用户同时是多个组中的成员时，在/etc/passwd 文件中记录的是用户所属的主组，也就是登录时所属的默认组，而其他组称为附加组。用户要访问属于附加组的文件时，必须首先使用 newgrp 命令使自己成为所要访问的组中的成员。用户组的所有信息都存放在/etc/group 文件中。此文件的格式也类似于/etc/passwd 文件，由冒号隔开若干个字段，这些字段有：

组名，口令，组标识号，组内用户列表。

①“组名”是用户组的名称，由字母或数字构成。与/etc/passwd 中的登录名一样，组名不应重复。

②“口令”字段存放的是用户组加密后的口令字。一般 Linux 系统的用户组都没有口令，即这个字段一般为空，或者是 * 。

③“组标识号”与用户标识号类似，也是一个整数，被系统内部用来标识组。

④“组内用户列表”是属于这个组的所有用户的列表，不同用户之间用逗号分隔。这个用户组可能是用户的主组，也可能是附加组。

/etc/group 文件的一个例子如下：

```
# cat /etc/group
root::0:root
bin::2:root,bin
sys::3:root,uucp
adm::4:root,adm
daemon::5:root,daemon
lp::7:root,lp
users::20:root,sam
```

4. 添加量用户批

添加和删除用户对每位 Linux 系统管理员都是轻而易举的事，比较棘手的是在添加几十

个、上百个甚至上千个用户时，我们不太可能还使用 useradd 一个一个地添加，必然要找一种简便的创建大量用户的方法。Linux 系统提供了创建大量用户的工具，可以立即创建大量用户，方法如下：

1）先编辑一个文本用户文件，每一列按照/etc/passwd 密码文件的格式书写，要注意每个用户的用户名、UID、宿主目录都不可以相同，其中密码栏可以留做空白或输入 x 号。一个范例文件 user. txt 内容如下：

```
user001::600:100:user:/home/user001:/bin/bash
user002::601:100:user:/home/user002:/bin/bash
user003::602:100:user:/home/user003:/bin/bash
user004::603:100:user:/home/user004:/bin/bash
user005::604:100:user:/home/user005:/bin/bash
user006::605:100:user:/home/user006:/bin/bash
```

2）以 root 身份执行命令/usr/sbin/newusers，从刚创建的用户文件 user. txt 中导入数据，创建用户：

```
# newusers  <  user. txt
```

然后可以执行命令 vipw 或 vi /etc/passwd 检查/etc/passwd 文件是否已经出现这些用户的数据中，并且用户的宿主目录是否已经创建。

3）执行命令/usr/sbin/pwunconv，将/etc/shadow 产生的 shadow 密码解码，然后回写到/etc/passwd 中，并将/etc/shadow 的 shadow 密码栏删掉。这是为了方便下一步的密码转换工作，即先取消 shadow password 功能。

4）编辑每个用户的密码对照文件，范例文件 passwd. txt 内容如下：

```
user001：密码
user002：密码
user003：密码
user004：密码
user005：密码
user006：密码
```

5）以 root 身份执行命令/usr/sbin/chpasswd，创建用户密码，chpasswd 会将经过/usr/bin/passwd 命令编码过的密码写入/etc/passwd 的密码栏。

6）确定密码经编码写入/etc/passwd 的密码栏后，执行命令/usr/sbin/pwconv 将密码编码为 shadow password，并将结果写入/etc/shadow。

这样就完成了大量用户的创建，之后可以到/home 下检查这些用户宿主目录的权限设置是否都正确，并登录验证用户密码是否正确。

5. 赋予普通用户特殊权限

在 Linux 系统中，管理员往往不止一人，若每位管理员都用 root 身份进行管理工作，根本无法弄清楚谁该做什么。所以最好的方式是：管理员创建一些普通用户，分配一部分系统管理工作给他们。

我们不可以使用 su 让他们直接变成 root，因为这些用户都必须知道 root 的密码，这种方法很不安全，而且也不符合我们的分工需求。一般的做法是利用权限的设置，依工作性质分

类，让特殊身份的用户成为同一个工作组，并设置工作组权限。例如：要 wwwadm 这位用户负责管理网站数据，一般 Apache Web Server 的进程 httpd 的所有者是 www，您可以设置用户 wwwadm 与 www 为同一工作组，并设置 Apache 默认存放网页目录/usr/local/httpd/htdocs 的工作组权限为可读、可写、可执行，这样属于此工作组的每位用户就可以进行网页的管理了。

但这并不是最好的解决办法，例如管理员想授予一个普通用户关机的权限，这时使用上述的办法就不是很理想。这时您也许会想，我只让这个用户可以以 root 身份执行 shutdown 命令就行了。完全没错，可惜在通常的 Linux 系统中无法实现这一功能，不过已经有工具可以实现这样的功能——sudo。

sudo 通过维护一个特权到用户名映射的数据库将特权分配给不同的用户，这些特权可由数据库中所列的一些不同的命令来识别。为了获得某一特权项，有资格的用户只需简单地在命令行输入 sudo 与命令名之后，按照提示再次输入口令（用户自己的口令，不是 root 用户口令）。例如，sudo 允许普通用户格式化磁盘，但是却没有赋予其他的 root 用户特权。

sudo 工具由文件/etc/sudoers 进行配置，该文件包含所有可以访问 sudo 工具的用户列表并定义了它们的特权。一个典型的/etc/sudoers 条目如下：

liming ALL = （ALL） ALL

这个条目使得用户 liming 作为超级用户访问所有应用程序，如用户 liming 需要作为超级用户运行命令，他只需简单地在命令前加上前缀 sudo。因此，要以 root 用户的身份执行命令 format，liming 可以输入如下命令：

sudo /usr/sbin/useradd sam

注意：命令要写绝对路径，/usr/sbin 默认不在普通用户的搜索路径中，或者加入此路径：PATH = $ PATH：/usr/sbin；export PATH。另外，不同系统命令的路径不尽相同，可以使用命令“whereis 命令名”来查找其路径。

这时会显示下面的输出结果：

We trust you have received the usual lecture from the local System

Administrator. It usually boils down to these two things：

#1） Respect the privacy of others.

#2） Think before you type.

Password：

如果 liming 正确地输入了口令，命令 useradd 将会以 root 用户身份执行。

注意：配置文件/etc/sudoers 必须使用命令 Visudo 来编辑。

只要把相应的用户名、主机名和许可的命令列表以标准的格式加入到文件/etc/sudoers，并保存就可以生效。

例子 1：管理员需要允许 gem 用户在主机 sun 上执行 reboot 和 shutdown 命令，在/etc/sudoers 中加入：

gem sun =/usr/sbin/reboot，/usr/sbin/shutdown

注意：命令一定要使用绝对路径，以避免其他目录的同名命令被执行，从而造成安全隐患。

然后保存退出，gem 用户想执行 reboot 命令时，只要在提示符下运行下列命令：

$ sudo /usr/sbin/reboot

输入正确的密码，就可以重启服务器了。

如果想对一组用户进行定义，可以在组名前加上%，对其进行设置，如：

%cuug ALL = （ALL） ALL

另外，还可以利用别名来简化配置文件。别名类似组的概念，有用户别名、主机别名和命令别名。多个用户可以首先用一个别名来定义，然后在规定他们可以执行什么命令的时候使用别名就可以了，这个配置对所有用户都生效。主机别名和命令别名也是如此。注意使用前先要在/etc/sudoers 中定义 User_Alias，Host_Alias，Cmnd_Alias 项，在其后面加入相应的名称，也以逗号分隔开就可以了，举例如下：

Host_Alias SERVER = no1

User_Alias ADMINS = liming，gem

Cmnd_Alias SHUTDOWN = /usr/sbin/halt，/usr/sbin/shutdown，/usr/sbin/reboot

ADMINS SERVER = SHUTDOWN

例子 2：

ADMINS ALL = （ALL） NOPASSWD：ALL

表示允许 ADMINS 不用口令执行一切操作，其中“NOPASSWD:”项定义了用户执行操作时不需要输入口令。

sudo 命令还可以加上一些参数，完成一些辅助的功能，如

$ sudo -l

会显示出类似这样的信息：

User liming may run the following commands on this host：

(root) /usr/sbin/reboot

说明 root 允许用户 liming 执行/usr/sbin/reboot 命令。这个参数可以使用户查看自己目前可以在 sudo 中执行哪些命令。

在命令提示符下键入 sudo 命令会列出所有参数，其他一些参数如下：

-V 显示版本编号；

-h 显示 sudo 命令的使用参数；

-v 因为 sudo 在第一次执行时或是在 *N* 分钟内没有执行（*N* 预设为 5）会询问密码。这个参数是重新做一次确认，如果超过 *N* 分钟，也会询问密码；

-k 将会强迫使用者在下一次执行 sudo 时询问密码（不论有没有超过 *N* 分钟）；

-b 将要执行的命令放在背景执行；

-p prompt 可以更改询问密码的提示语，其中%u 会替换为使用者的账号名称，%h 会显示主机名称；

-u username/#uid 不加此参数，代表要以 root 的身份执行命令，而加了此参数，可以以 username 的身份执行命令（#uid 为该 username 的 UID）；

-s 执行环境变量中的 SHELL 所指定的 Shell ，或是 /etc/passwd 里所指定的 Shell；

-H 将环境变量中的 HOME（宿主目录）指定为要变更身份的使用者的宿主目录（如不加-u 参数就是系统管理者 root）。

3.2 企业项目场景 3：Linux 系统用户管理

在安装完服务器后，宁波四海进出口公司需要在公司内设置服务器使用的用户账户。账户需要按部门分组建立，具体要求如下：

表 3-1 部门标识和用户标识要求

部门名称	组标识	用户开头标识	用户数
总经理办公室	ZJL	ZJL_	3
财务部	CWB	CWB_	3
后勤部	HQB	HQB_	3
业务 1 部	YW1	YW1_	5
业务 2 部	YW2	YW2_	5

其他要求：

1）建立用户时，需将部门名称写入注释，并设定有效日期；

2）可根据需要暂停或启用某用户；

3）在服务器中应设置两个以上超级用户，并对超级用户的密码丢失采取恢复措施。

3.3 Linux 系统用户管理教学目标（8 学时）

1. 教学目标

最终目标：完成用户的添加、删除、停用、启用和属性管理。

促成目标：

1）掌握用户添加删除基本方法；

2）掌握组的添加删除方法；

3）知道用户和组的文档所在目录和文件名；

4）能熟练修改用户属性；

5）能熟练将用户添加到指定组；

6）能按要求停用、启用用户或指定用户有效日期；

7）能根据文件成批添加用户。

2. 工作任务

1）添加指定组和用户；

2）将用户添加到指定组；

3）修改用户口令；

4）指定用户有效日期、用户目录、注释、默认使用 Shell 等属性；

5）根据文件成批添加用户，并修改用户口令；

6）停用、启用指定用户。

3. 活动设计

添加指定组和用户，规定用户组、有效日期、用户目录、注释、使用 Shell 内容。

改变指定用户的属性。

停用、启用指定用户。

3.4 项目三　实验指导

3.4.1 新建组

```
#groupadd  组名              ；#代表超级用户的提示符，普通用户提示符 $
#groupadd  -g 组编号 组名   ；可以指定编号，但必须保证唯一性
```

例如：

```
#groupadd group1             ；建立 group1 组
#cat  /etc/group             ；查看组档案
```

3.4.2 删除组

```
#groupdel 组名
```

例如：

```
#groupdel group1             ；删除组 group1
#cat  /etc/group             ；查看组档案
```

注意：需删除的组不能含有组成员。

3.4.3 新建用户

useradd 功能说明：建立用户账号。

语　法：useradd［-m Mnr］［-c <备注>］［-d <登入目录>］［-e <有效期限>］［-f <缓冲天数>］［-g <群组>］［-G <群组>］［-s <shell>］［-u <uid>］［用户账号］或 useradd -D［-b］［-e <有效期限>］［-f <缓冲天数>］［-g <群组>］［-G <群组>］［-s <shell>］

说明：useradd 可用来建立用户账号。账号建好之后，再用 passwd 设定账号的密码，可用 userdel 删除账号。使用 useradd 指令所建立的账号，实际上是保存在/etc/passwd 文本文件中。

参　数：

-c　<备注>　加上备注文字。备注文字会保存在 passwd 的备注栏位中。

-d　<登入目录>　指定用户登入时的启始目录。

-D　变更预设值。

-e　<有效期限>　指定账号的有效期限。

-f　<缓冲天数>　指定在密码过期后多少天即关闭该账号。

-g　<群组>　指定用户所属的群组。

-G　<群组>　指定用户所属的附加群组。

-m　自动建立用户的登入目录。

-M　不要自动建立用户的登入目录。

-n　取消建立以用户名称为名的群组。

-r 建立系统账号。
-s <shell> 指定用户登入后所使用的 shell。
-u <uid> 指定用户 ID。

```
#useradd 用户名                 ; 最简单的新建用户
#useradd -g 组名 用户名         ; 新建用户并加入组
```

完整的新建用户参数：

```
#useradd -g 组名-d 目录-s shell 类型-e 有效期-c 注释-u 用户编号用户名
```

例如：

```
#useradd user1                  ; 新建用户 user1
#useradd -g group1 user2        ; 新建用户 user2，并加入 group1 组
#useradd -g group1 user3-d/usr/user3-s/bin/sh-e 2009/12/31-c 测试用户 user3
#passwd 用户名                  ; 修改用户口令
 $ passwd                       ; 修改用户自己的口令
#cat /etc/passwd                ; 查看用户档案，检查用户是否加入
```

3.4.4 删除用户

```
#userdel-r 用户名               ; 加参数-r，连同用户目录删除
```

例如：

```
#userdel user1                  ; 删除 user1 用户，目录未删除
#userdel-r user2                ; 完全删除 user2 用户
```

3.4.5 修改用户属性

```
#usermod -g 组名-d 目录-s shell 类型-e 有效期-c 注释-u 用户编号 用户名
```

例如：

```
#usermod -e 2010-1-1-c 测试用户 user2    ; 修改 user2 的有效期和注释
```

#usermod -s/bin/false user3 ；修改 user3 的 shell 为无效（不存在），这样 user3 是不能登录的，但可以有其他功能，如发送邮件，访问共享资源等。

3.4.6 暂停和恢复用户账号

```
#passwd -l 用户名            ; 暂停用户账号
#passwd -u 用户名            ; 恢复用户账号
```

3.4.7 超级用户管理（学习 Vi 编辑后再做）

1. 新增超级用户账户

增加一个普通账户，将组编号和用户编号都改为 0（需要修改/etc/passwd 文件）。

2. 超级用户口令丢失处理

在启动 Linux 系统时，进入 GRUB 菜单，编辑启动命令，使系统直接进入单用户状态，然后修改口令即可。编辑方法如下：

1）启动时在规定时间内，输入回车键，进入 GRUB 菜单；

2）按 e（edit）进入下层菜单；

3）选择第 2 项按 e 编辑，在单词 ro 前加入 single，回车返回；

Kernel/vmlinuz-2.4.20-8 single ro root = LABEL = /

4）光标仍在第 2 项，按 b(boot）启动 Linux，进入单用户；

5）在单用户下修改口令。

#passwd 用户名

#init 5 ；直接进入图形多用户

3.5 项目三完成步骤详解

3.5.1 新建组

1）新建总经理办公室组 ZJL、CWB、HQB、YW1 和 YW2；

groupadd ZJL

groupadd CWB

groupadd HQB

groupadd YW1

groupadd YW2

添加组命令如图 3-1 所示。

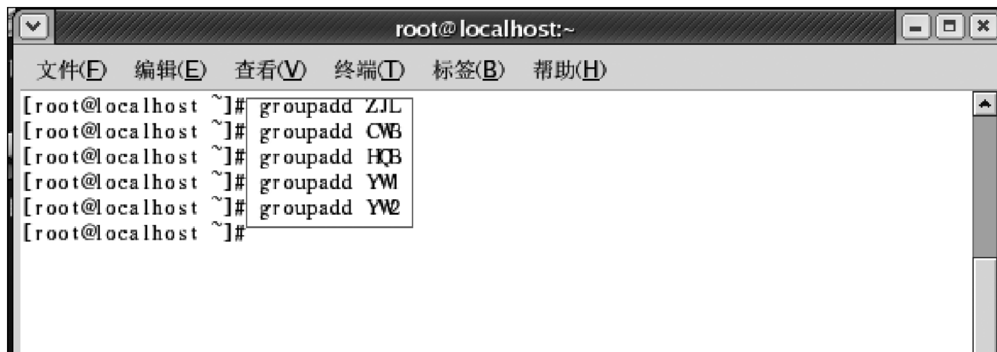


图 3-1 添加组命令

2）查看新建组信息；

#cat /etc/group

显示组文档如图 3-2 所示。

3.5.2 新建用户

1）按照要求新建总经理办公室组的 3 个用户 ZJL_user1、ZJL_user2、ZJL_user3，注释为“总经理办公室”，设定有效期到 2011 年 04 月 30 日（见图 3-3）；

2）按照要求新建财务部组的 3 个用户 CWB_user1、CWB_user2、CWB_user3，注释为“财务部”，设定有效期到 2011 年 04 月 30 日（见图 3-4）；

```
xfs:x:43:
ntp:x:38:
gdm:x:42:
htt:x:101:
ZJL:x:500:
CWB:x:501:
HQB:x:502:
YW1:x:503:
YW2:x:504:
[root@localhost ~]#
```

图 3-2　显示组文档

```
root@localhost:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(B)  帮助(H)
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user1
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user2
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user3
[root@localhost ~]#
```

图 3-3　新建用户命令

```
root@localhost:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(B)  帮助(H)
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user1
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user2
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user3
[root@localhost ~]# useradd -g CWB -e 2011-04-30 -c 财务部 CWB_user1
[root@localhost ~]# useradd -g CWB -e 2011-04-30 -c 财务部 CWB_user2
[root@localhost ~]# useradd -g CWB -e 2011-04-30 -c 财务部 CWB_user3
[root@localhost ~]#
```

图 3-4　新建用户命令

3）按照要求新建后勤部组的 3 个用户 HQB_user1、HQB_user2、HQB_user3，注释为“后勤部”，设定有效期到 2011 年 04 月 30 日（见图 3-5）；

```
root@localhost:~
文件(F)  编辑(E)  查看(V)  终端(T)  标签(B)  帮助(H)
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user1
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user2
[root@localhost ~]# useradd -g ZJL -e 2011-04-30 -c 总经理办公室 ZJL_user3
[root@localhost ~]# useradd -g CWB -e 2011-04-30 -c 财务部 CWB_user1
[root@localhost ~]# useradd -g CWB -e 2011-04-30 -c 财务部 CWB_user2
[root@localhost ~]# useradd -g CWB -e 2011-04-30 -c 财务部 CWB_user3
[root@localhost ~]# useradd -g HQB -e 2011-04-30 -c 后勤部 HQB_user1
[root@localhost ~]# useradd -g HQB -e 2011-04-30 -c 后勤部 HQB_user2
[root@localhost ~]# useradd -g HQB -e 2011-04-30 -c 后勤部 HQB_user3
[root@localhost ~]#
```

图 3-5　新建用户命令

4）按照要求新建业务 1 部组的 5 个用户 YW1_user1、YW1_user2、YW1_user3、YW1_user4、YW1_user5，注释为“业务 1 部”，设定有效期到 2010 年 12 月 31 日（见图 3-6）；

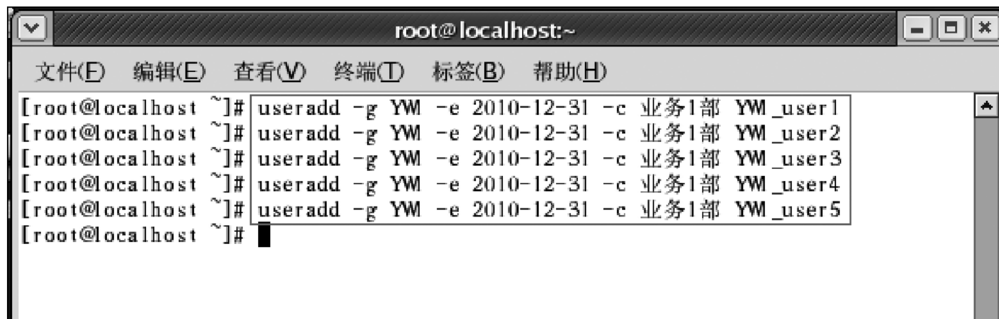


图 3-6　新建用户命令

5）按照要求新建后勤部组的 5 个用户 YW2_user1、YW2_user2、YW2_user3、YW2_user4、YW2_user5，注释为“业务 2 部”，设定有效期到 2010 年 12 月 31 日（见图 3-7）；

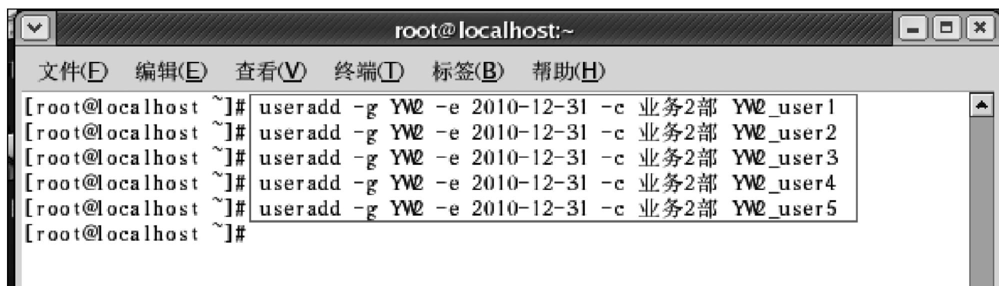


图 3-7　新建用户命令

6）查看用户信息。

cat /etc/passwd

显示用户文档如图 3-8 所示。

3.5.3　暂停和恢复用户账号

#passwd -l YW1_user1　　；暂停用户账号 YW1_user1

暂停用户命令如图 3-9 所示。

#passwd -u YW1_user1　　；恢复用户账号 YW1_user1

恢复用户命令如图 3-10 所示。

3.5.4　超级用户管理（学习 Vi 编辑后再做）

1. 新增超级用户账户

将总经理办公室的 3 个普通账户 ZJL_user1、ZJL_user2、ZJL_user3 都设为超级用户，将这些用户的组编号和用户编号都改为 0（需要修改/etc/passwd 文件）。

#vi /etc/passwd

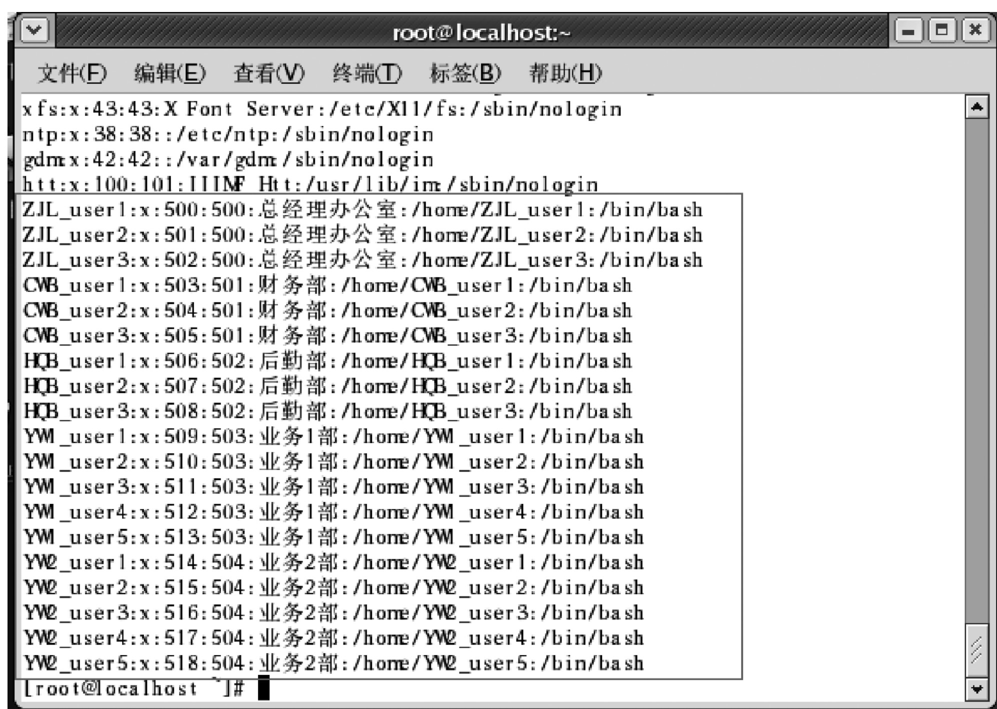


图 3-8　显示用户文档



图 3-9　暂停用户命令

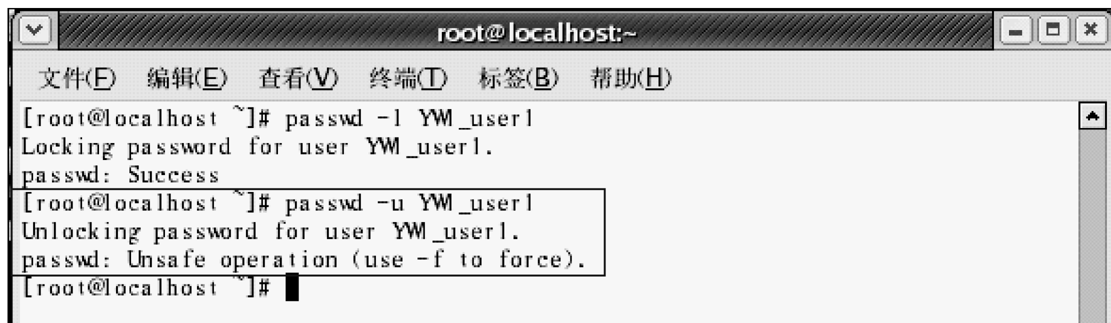


图 3-10　恢复用户命令

在编辑模式下将 ZJL_user1、ZJL_user2、ZJL_user3 3 个用户的组编号和用户编号修改为 0 后，保存退出 vi。

编辑用户文档如图 3-11 所示。

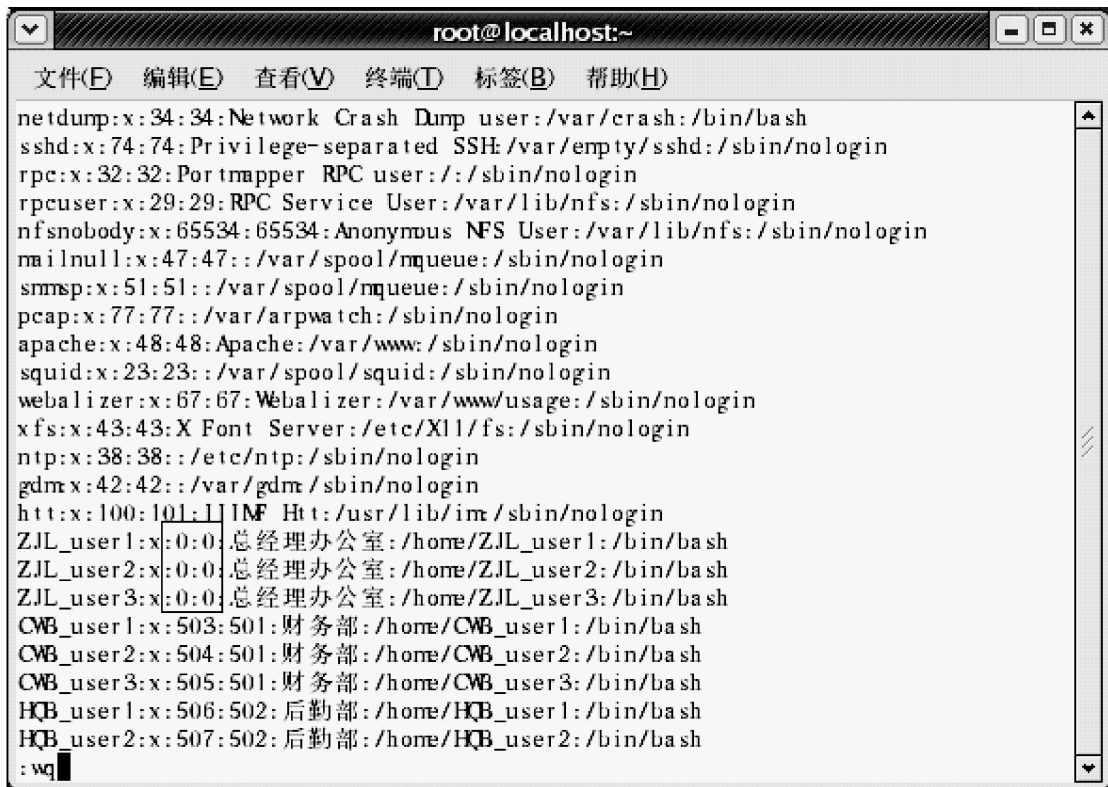


图 3-11 编辑用户文档

2. 超级用户口令丢失处理

在启动 Linux 系统时，进入 GRUB 菜单，编辑启动命令，使系统直接进入单用户状态，然后修改口令即可。编辑方法如下：

1）启动时在规定时间内，输入回车键，进入 GRUB 菜单。

系统启动界面如图 3-12 所示。

2）按 e(edit）进入下层菜单。

进入编辑命令如图 3-13 所示。

3）选择第 2 项按 e 编辑，在单词 ro 前加入 single，回车返回。

Kernel/vmlinuz-2.4.20-8 single ro root = LABEL = /

修改启动命令如图 3-14 所示。

4）光标仍在第 2 项，按 b(boot）启动 Linux，进入单用户，在单用户下修改口令。

#passwd 用户名（这里以 root 用户为例）

修改超级用户口令如图 3-15 所示。

#init 5 ；直接进入图形多用户



图 3-12　系统启动界面



图 3-13　进入编辑命令

重启系统至图形界面如图 3-16 所示。

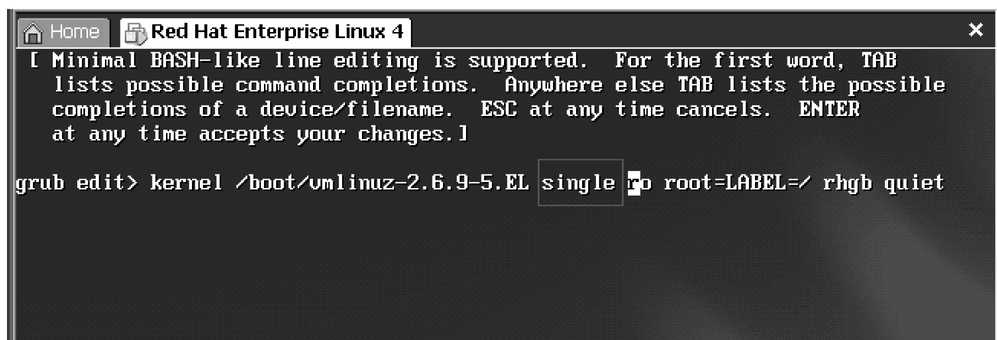


图 3-14　修改启动命令

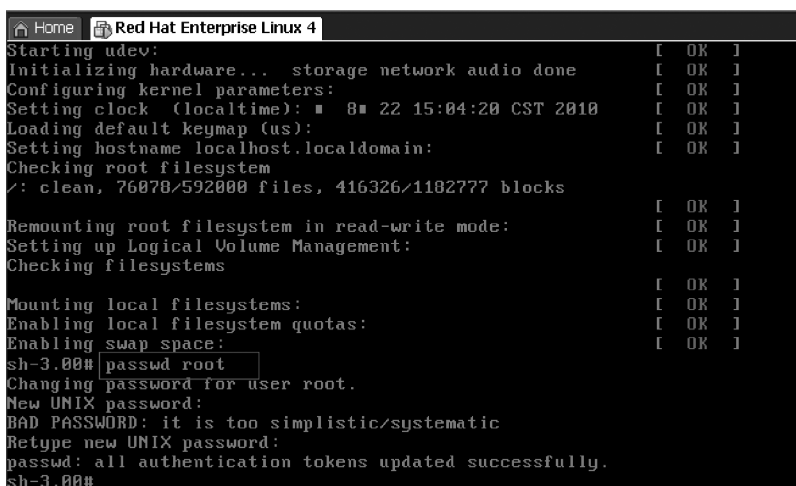


图 3-15　修改超级用户口令

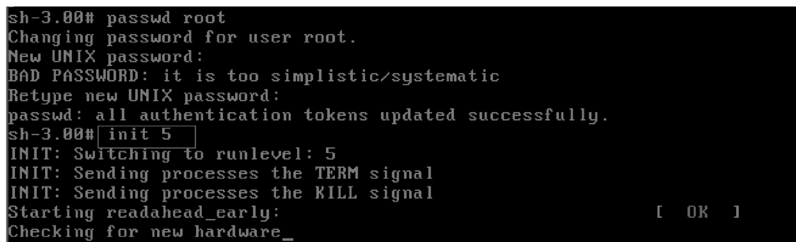


图 3-16　重启系统至图形界面

3.6　思考题

1）超级用户和普通用户的提示符有什么不同？
2）普通用户的默认的主目录在哪个目录下？
3）用户的主目录可以更改么？如果可以如何更改？
4）用户密码丢失怎么处理？

第4讲 Linux 文件系统管理



导读

文件和目录是操作系统中十分重要的概念，文件和目录实现了操作系统对系统和用户的数据的管理，了解文件和目录的有关概念是理解操作系统的一个重要部分。

4.1 引言

4.1.1 文件和目录概述

1. 文件

绝大多数操作系统（从 DOS 到 Windows，从 UNIX 到 Linux）中都有文件的概念。简单地讲文件就是一组相关纪录的集合。Linux 中文件是无结构的字符流，即文件中的任两个字节是完全独立的。具体的信息要放到相应的环境中才能理解。

文件通过文件名来标识引用，即文件是通过文件名来进行管理的。系统中每一个文件都有一个文件名。

2. 目录

文件多了就会发生混乱，因此目录就出现了。目录就是存放一组文件的“夹子”，Windows 中的“文件夹”就是这个概念。所以目录就是一组相关文件的集合，我们通常都通过目录来管理文件。目录和文件一样也有自己的名字。而一个目录下面除了可以放文件之外，还可以存放目录，称为这个目录的子目录。这个子目录之下还可以有它自己的子目录，依此类推从而形成一个树状目录结构。其中，最上层的目录我们称为根目录，而系统内的所有目录都是根目录的子目录。

3. 路径

相信使用过其他操作系统的用户对这个概念不会陌生。打个比方，某个文件就好像是在一个写字楼中的一间办公室内的一把椅子，当然还有其他很多椅子也在这个办公室里。现在要把它找出来坐在上面办公，那么只知道这个椅子是什么样还不行，还需要知道如何到达哪里。文件名就是这椅子，而路径就是用来说明如何找到该椅子的。

Linux 文件系统是由目录和目录下的文件一起构成的，是一个树形的结构。在这个系统最顶层的是根目录“/”。下面是各级其他目录。Linux 路径是由从根目录到文件所经过的所

有目录加上文件名连在一起构成的，中间以“/”分隔。同样的和 DOS 相同，在每个目录上面都有名为“.”和“..”的两个文件，前者代表的是当前目录，后者代表的是当前目录的父目录（上一级目录）。如果要存取不在工作目录中的文件，则除了文件名之外，也要指定目录的位置，这就是路径。路径由文件所在的目录加上文件名组成，中间再用“/”隔开。系统中每一个文件都有一个路径名，要想引用这个文件就要通过路径来引用。

路径又分成以下两种：

绝对路径：路径名称以“/”开头，其中文件或目录的位置从根目录算起。

相对路径：路径名称不以“/”开头，其中文件或目录的位置从工作目录算起，工作目录可以理解为当前所在的目录。

4. 文件目录命名规则

和 DOS 相比，L 文件命名相当宽松。L 文件名可以由字母、数字和一些标点符号组成。文件名中不能包含有空格和下列的字符：

！ @ # $ % ^& * ( ) " \ / | ； < >

但文件及目录名字的长度可以超过 DOS 的 8.3 的规定，只是通常要在 256 字之内。为了兼容 UNIX 文件系统，建议大家使用 14 个以下的字符作为文件名（一般的 UNIX 系统都只允许 14 个以下的字符作为文件名）。另外，我们大家经常使用的 Windows 95/98 系统是可以在文件名中使用空格的，Linux 系统实际上也有办法可以解决这个问题，那就是将文件名使用“”括起来，这样在双引号里面就可以随意使用任何字符了。但除非是真有必要，一般建议最好不要这么做。因为大多数程序不一定可以访问具有这种文件名的文件，而且这样使得文件系统的可移植性也变差。

5. 处理文件的命令（见表 4-1）

表 4-1 文件操作常用命令

命　令	作　用
ls	列出目录的内容
ls -l	列出目录的详细内容
cd < dir >	切换工作目录为 < dir >
Mkdir < dir >	建立名叫 < dir > 的目录
Rmdir < dir >	删除名叫 < dir > 的目录
cat < file >	看 < file > 的内容
More < file >	看 < file > 的内容，并且每印满一页后则暂停
rm < file >	删除 < file >
cp < file-A > < file-B >	把 < file-A > 复制一份成 < file-B >
ev < file-A > < file-B >	将 < file-A > 改名为 < file-B >
Echo < string >	将 < string > 显示在屏幕上

4.1.2 文件类型

Linux 文件系统中包括以下类型的文件：

文本文件：同其他操作系统的文本文件相同，是由一些字符在计算机内的 ASCII 码组成

的。通常这类文件在各个操作系统中是兼容的。

二进制文件：文件中保存的是数据的二进制表示，一般情况下用普通的文本阅览器所见到的是一堆乱码，需要专用的软件才能查看文件内容。例如：可执行文件、图像文件、声音文件等都是这类文件。

目录文件：Linux 中对目录的管理也是通过文件进行的。顾名思义，目录文件就是一类特殊的文件，其中保存的信息不是普通的数据，而是文件名和文件有关的文件信息，例如文件名、文件创建日期、文件类型、读取权限等。目录文件中还可以包含下一级的目录。

连接文件：普通的连接实际上不是文件，它们仅是指向同一索引节点的目录条目，是一个索引节点表。该表记录了一个文件有多少连接，这种连接不能跨越设备，因为不同设备上的 i 节点号不同，而另一种连接类型—符号链接，则没有这种限制。

设备文件：设备文件是 Linux 系统中较特殊的文件。Linux 中引入设备文件的目的是实现设备独立性。用户访问外部设备时就是通过设备文件进行的。操作系统对外设的支持程度直接影响到操作系统被使用的广泛程度，一个由于本身而限制了访问硬件的操作系统不能说是一个成功的操作系统。面对越来越多、越来越高级的外设，Linux 将外设看作一个文件来管理，这样就避免了由于外设的增加而带来的问题。当需要增加新设备时，只要在操作系统内核中增加相应的设备文件即可。因此，设备文件在外设与操作系统之间提供了一个接口，用户使用外设就像使用普通文件一样。设备文件存放在/dev 目录下，它使用设备的主设备号和次设备号来区分指定的外设：主设备号说明设备类型，次设备号说明具体指哪一个设备。例如：/dev/fd0 指系统中的一个软盘驱动器，主设备号 fd 是软盘驱动器（floppydisk）的缩写；次设备号 0 是软盘驱动器编号，意思是指系统中的一软盘驱动器，也就是我们通常所说的“A：”盘。

管道文件：也是 Linux 中较特殊的文件类型，这类文件多用于进程间的通信方面。进程可通过管道文件实现互相的信息传输，一个进程将数据写在管道的一端，另一个进程从管道的另一端将数据读出，这样就能实现进程的相互协作。管道文件又可分为无名管道和有名管道两种。

4.1.3 文件系统和标准文件系统布局

操作系统通过文件系统管理系统中的数据，Linux 与 UNIX 的文件系统布局基本相同，都是由根目录和子目录构成的。

1. 文件系统

文件系统是操作系统中非常重要的概念。文件系统管理机器上的文件目录，使之有效地被存取访问。文件系统是操作系统用于明确磁盘或分区上的文件的方法和数据结构，即在磁盘上组织文件的方法。也可以指用于存储文件的磁盘或分区，或文件系统种类。简单地讲，文件系统是指按照一定规则组织的文件结构。Linux 中文件系统是按照树形结构组织的，所有 Linux 的数据（包括程序、库、用户文件等）都是由文件系统按照树形目录结构管理的。这些概念和 DOS/Windows 是一样的。

Linux 文件系统的树形结构如图 4-1 所示。其中，每台机器都有根文件系统（一般在本地盘中，当然也可以在 RAM 盘或网络盘中）。它包含系统引导和使其他文件系统得以 mount 所必要的文件，根文件系统应该有单用户状态所必需的足够的内容，还应该包括修复损坏系

统、恢复备份等的工具。

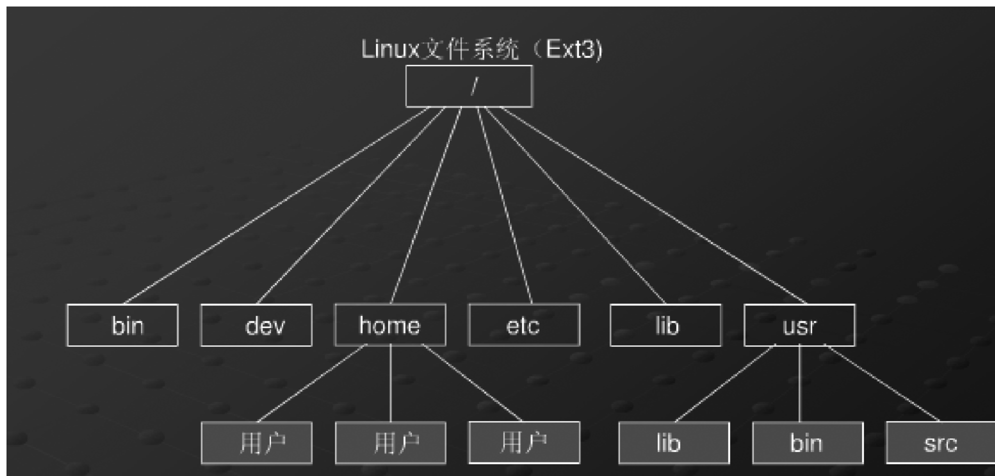


图 4-1 Linux 文件系统的树形结构

除了根文件系统外，用户可以根据自己的需要创建一个或多个其他的文件系统。

2. 标准文件系统布局

标准文件系统布局将整个目录树分为几个小的部分，每个部分可以在自己的磁盘或分区上，只要能为磁盘容量所容纳，并易于备份及其他系统管理即可。文件系统主要部分是根、/usr、/var 和/home 文件系统。每个部分各有其不同的目的。目录树已被设计成能在 Linux 系统的网络中很好地工作，可以通过只读设备（如 CD – ROM）或 NFS 网络共享文件系统的一些部分。

下面对几个重要的文件系统做一下简要介绍：

（1）根文件系统

根文件系统是每台计算机都必须具有的部分。它包含系统引导和使其他文件系统得以 mount 所必要的文件。根文件系统应该有单用户状态所必需的足够的内容，还应该包括修复损坏系统、恢复备份等的工具。根文件系统一般应该比较小，因为包括严格的文件和一个小的不经常改变的文件系统通常都不容易损坏。如果根文件系统发生损坏，则意味着要用特定的方法（例如从软盘）引导系统，否则系统将无法引导，所以要格外小心。

（2）/usr 文件系统

/usr 文件系统包含所有命令、库、帮助（man）页和其他一般操作中所需的不改变的文件。/usr 不应该包括对特定机器使用的特定的文件，也不应该有一般使用中要修改的文件。这样允许此文件系统中的文件通过网络共享，才可以使之更有效，因为这样节省了磁盘空间（/usr 很容易是数百兆），且易于管理（当升级应用时，只有主/usr 需要改变，而无须改变每台机器）。即使此文件系统在本地盘上，也可以将其设为只读属性，以减少系统崩溃时对文件系统的损坏。

（3）/var 文件系统

/var 文件系统包含会改变的文件，比如 spool 目录（mail、news、打印机等用的），log（日志）文件、formatted manual pages 和暂存文件。传统上/var 的所有东西曾在/usr 下的某

个地方，但这样/usr 就不可能只读安装了。

（4）/home 文件系统

/home 文件系统包含用户本地目录，即系统上的所有实际数据。最好将/home 目录安装到专门的目录树或文件系统（分区）中，因为这样易于备份，而不必备份其他文件系统（至少不必经常备份，因为它们很少改变）。一个大的/home 可能要分为若干文件系统，需要在/home 下加一级名字，如/home/students、/home/staff 等。

虽然上面将不同的部分称为文件系统，但并不需要将它们安装在不同的磁盘上，它们也不必是真的分离的文件系统。如果用户系统是小的单用户系统，并且用户希望使它简单化，那么它们可以很容易地放在一个文件系统中。根据磁盘容量和不同目的所需分配的空间，目录树也可以分到不同的文件系统中。重要的是使用标准的名字，即使/var 和/usr 在同一分区上，名字/usr/lib/libc. a 和/var/adm/messages 必须能工作，例如将/var 下的文件移动到/usr/var，并将/var 作为/usr/var 的符号连接，这样使文件组织有序，并且更加容易查找。另一个方法是根据属于的程序分组文件，即所有 Emacs 文件在一个目录中，所有 txt 文件在另一个中，等等。后一种方法的问题是文件难于共享（程序目录经常同时包含静态可共享的和动态不可共享的文件），有时难于查找。

3. 根文件系统目录介绍

通常情况下，根文件系统所占空间一般应该比较小，因为其中的绝大部分文件都不需要经常改动，而且包括严格的文件和一个小的不经常改变的文件系统不容易损坏。除了可能的一个叫/vmlinuz 标准的系统引导映像之外，根目录一般不含任何文件。所有其他文件在根文件系统的子目录中。

（1）/bin 目录

/bin 目录包含了引导启动所需的命令或普通用户可能用的命令（可能在引导启动后）。这些命令都是二进制文件的可执行程序（bin 是 binary--二进制的简称），多是系统中重要的系统文件。

（2）/sbin 目录

/sbin 目录类似/bin，也用于存储二进制文件。因为其中的大部分文件多是系统管理员使用的基本的系统程序，所以虽然普通用户必要且允许时可以使用，但一般不给普通用户使用。

（3）/etc 目录

/etc 目录存放着各种系统配置文件，其中包括了用户信息文件/etc/passwd、系统初始化文件/etc/rc 等。Linux 正是靠这些文件才得以正常地运行。

（4）/root 目录

/root 目录是超级用户的目录。

（5）/lib 目录

/lib 目录是根文件系统上的程序所需的共享库，存放了根文件系统程序运行所需的共享文件。这些文件包含了可被许多程序共享的代码，以避免每个程序都包含有相同的子程序的副本，故可以使得可执行文件变得更小，节省空间。

（6）/lib/modules 目录

/lib/modules 目录包含系统核心可加载各种模块，尤其是那些在恢复损坏的系统时重新

引导系统所需的模块（例如网络和文件系统驱动）。

（7）/dev 目录

/dev 目录存放了设备文件，即设备驱动程序，用户通过这些文件访问外部设备。比如，用户可以通过访问/dev/mouse 来访问鼠标的输入，就像访问其他文件一样。

（8）/tmp 目录

/tmp 目录存放程序在运行时产生的信息和数据。但在引导启动后，运行的程序最好使用/var/tmp 来代替/tmp，因为前者可能拥有一个更大的磁盘空间。

（9）/boot 目录

/boot 目录存放引导加载器（bootstra ploader）使用的文件，如 LILO，核心映像也经常放在这里，而不是放在根目录中。但是如果有许多核心映像，这个目录就可能变得很大，这时使用单独的文件系统会更好一些。还有一点要注意的是，要确保核心映像必须在 IDE 硬盘的前 1024 柱面内。

（10）/mnt 目录

/mnt 目录是系统管理员临时安装（mount）文件系统的安装点。程序并不自动支持安装到/mnt。/mnt 下面可以分为许多子目录，例如/mnt /dosa 可能是使用 MSDOS 文件系统的软驱，而/mnt /exta 可能是使用 ext2 文件系统的软驱，/mnt /cdrom 光驱等。

（11）/proc、/usr、/var、/home 目录

其他文件系统的安装点。

4.2 企业项目场景 4：Linux 文件系统管理

在用户目录下需要根据业务需要建立文档和目录，并在本部门（本组）用户之间实现共享，有些目录和文档也需要在全公司范围共享，共享可分为只读和读写两种。现以财务部为例提出具体要求：

财务部成员目录和文档

表 4-2 共享文档权限要求

目录说明	目录标识	权限	文档
凭证汇总	Pz	本部门可读可写，其他部门无权使用	Pz1. txt Pz2. txt Pz3. txt
报表汇总	Bb	本部门可读可写，其他部门可读	Bb1. txt Bb2. txt
通知汇总	Tz	本部门可读可写，其他部门可读	Tz1. txt Tz2. txt
其他	Qt	全公司可读可写	Qt. txt

其他要求：

1）凭证文件在财务部是统一的，即财务部所有的人在编辑同一个文件，不需要互相

复制。

2）财务人员需要经常通过命令查找文件或文件中相关的内容，以便了解情况。

3）财务部需要对文件目录进行打包压缩，以便备份。并能在文件丢失或损坏的情况下从备份中恢复文件。

4.3　Linux 文件系统管理教学目标（16 学时）

1. 教学目标

最终目标：完成 Linux 系统文件的建立和管理。

促成目标：

1）熟悉文件和文件夹建立、删除、移动和改名、复制命令；

2）熟悉文件和文件夹查找、内容查找的命令使用；

3）掌握文件和文件夹存取权限属性修改方法；

4）掌握文件和文件夹压缩和解压缩。

2. 工作任务

1）按照要求建立文件夹，复制文件，移动文件；

2）按照要求改变文件夹的设置，删除部分文件和文件夹；

3）按照要求查找文件和文件内容；

4）按照要求设置文件和文件夹的存取权限；

5）按照要求压缩和解压缩文件和文件夹。

3. 活动设计

完成一个规定的文件系统设置。

4. 相关理论知识

5. 相关实践知识

利用文件夹压缩技术，做系统的备份和恢复。

4.4　项目四　实验指导

4.4.1　文件处理

1. 文件建立

```
$ cat > 文件名                ；用 cat 通过控制台输入建立一个文件，按 Ctrl + d 键结束
$ touch 文件名                ；建立一个空文件，可以一次建立多个文件，例如：
$ touch f1. txt f2. txt f3. txt
```

2. 复制文件

```
$ cp f1. txt f2. txt          ；将 f1. txt 复制为 f2. txt
```

3. 文件改名或移动

```
$ mv f1. txt f3. txt          ；将 f1. txt 改名为 f2. txt
```

4. 删除文件

```
$ rm   f1.txt          ; 删除文件 f1.txt
$ rm    *              ; 删除当前目录下所有文件
```

5. 文件列表（显示）

```
$ ls                   ; 列出所有文件和目录名（不包括隐含文件）
$ ls   -l              ; 详细列表
$ ls   -al             ; 列出隐含文件
$ ls *.txt             ; 列出 txt 扩展名文件
```

4.4.2 目录（文件夹）处理

1. 目录建立

```
$ mkdir 目录名              ; 可以一次建立多个目录
$ mkdir a b c d            ; 建立目录 a b c d 四个目录
$ mkdir a/a1 a/a1/a2       ; 分别在 a 目录下建立子目录 a1，a1 下再建 a2
$ mkdir   -p e/e1/e2/e3    ; 采用-p 参数，一次建立 4 层目录
```

2. 目录删除

```
$ rmdir d                  ; 删除目录 d，该目录必须为空，否则不能删除
$ rm   -r a                ; 删除 a 目录，a 目录可以不为空
```

3. 目录改名或移动（方法同文件处理）

```
$ mv e f                   ; 将目录 e 改为 f
$ mv b c                   ; 将目录 b 移动到 c 目录下（c 目录应存在）
$ mv f1.txt c/b            ; 将文件 f1.txt 移动到 c/b 目录下
```

4. 当前目录的切换

```
$ pwd                      ; 显示当前目录（路径）
$ mkdir   -p a/a1/a2/a3    ; 建立一个 4 层目录
$ cd a                     ; 切换到 a 目录，可以用 pwd 命令验证
$ cd a1                    ; 继续切换到 a/a1
$ cd a2/a3                 ; 切换到 a/a1/a2/a3
$ cd   ..                  ; 回退到 a/a1/a2
$ cd   ../../..            ; 回退 3 层到用户目录，若用户是 user1，默认用户目
                             录是/home/user1
$ cd a/a1/a2/a3            ; 一步切换到 a/a1/a2/a3
$ pwd                      ; 用 pwd 显示当前目录，应为/home/user1/a/a1/a2/a3
$ cd                       ; 返回用户根目录，根据用户而异
$ pwd                      ; 用 pwd 显示，若用户为 user1，应为/home/user1
```

5. 相对路径和绝对路径

若已有两个目录 a/a1/a2 和 b/b1/b2

```
$ touch f1.txt f2.txt      ; 建立两个文件 f1.txt 和 f2.txt
$ mv f1.txt a/a1           ; 将 f1.txt 文件移动到 a/a1 目录
```

$ mv f2. txt a/a1/a2　　　　　　　; 将 f2. txt 文件移动到 a/a1/a2 目录
$ cd b/b1　　　　　　　　　　　　; 将当前目录切换至 b/b1，用 pwd 检查
$ cp　/home/user1/a/a1/f1. txt　./ ; 采用绝对路径将 f1. txt 复制到当前目录，./代表当前目录
$ cp　../../a/a1/a2/f2. txt　./ ; 采用相对路径将 f2. txt 复制到当前目录

4.4.3　文件和目录的存取权限

1. 显示文件或目录的存取权限

例如，列表显示如下：

$ ls　-l　f1. txt　　　　　　　　; 详细列表 f1. txt 文件，看列表头部项目
-rw-r—r　--　1　　user1 group1　0　　10 月 20　08：40　f1. txt
存取权限　连接数　　用户组　　　大小　　日期　　时间　　文件名
$ mkdir a
$ ll　　　　　　　　　　　　　　; 详细列表，相当于 ls　-l
drwxr　-xr　-x　2　　user1　group1　4096　10 月 20　08：45　　a
-rw-r—r　--　1　　user1　group1　　0　10 月 20　08：40　f1. txt

可以看出，目录的第一个字母是 d，文件的第一个字母是-。还有其他类型的文件，这里暂不讨论。

第一个字母后面分为 3 段，分别代表用户（u）、同组（g）和其他用户（o）的权限，每段有 3 个权限：读 r、写 w、执行 x，没有权限用-表示。

2. 存取权限设置（文件和目录设置方法相同）

$ chmod u = rwx，g = rw，o = r f1. txt　　; 设置 f1. txt 用户的权限为可读、可写、可执行（rwx），同组用户的权限为可读、可写（rw-），其他用户权限为可读（r　--）
$ chmod o + x f1. txt　　　　　　　; 给 f1. txt 文件的其他用户增加可执行权限
$ chmod g-w f1. txt　　　　　　　　; 去掉同组用户可写的权限

采用加、减权限（+、-）的方法，其他权限不变，使用等号（=）将重新设置所有权限。

$ chmod u-x，g + w，o =　f1. txt　　; 用户和同组用户在原基础上增减，其他用户重新设置权限为空（没有权限），若有的段没有出现即不变
$ chmod 777 a　　　　　　　　　　; 用数字方法设置权限，每个权限用 0 或 1 表示，每段用八进制表示，最小 000 即 0，最大 111 即 7
$ chmod 764 f1. txt　　　　　　　　; 将 f1. txt 设置为 rwxrw-r　--，即 111 110 100

3. 用户文件共享设置

用户文件对外共享需要做两件事：

1）设置需要共享的文件本身权限；

2）设置文件所在目录的权限，使得其他用户能够进的来。

例如，要共享 user1 用户的 f1. txt 文件，操作如下：

$ chmod a = rw f1. txt　　　　　　; a 代表所有用户，即所有用户可读、可写，也可以写成 ugo = rwx
$ chmod 777　/home/user1　　　　; 设置用户 user1 的目录对所有用户开放，也

可以写成 a = rwx。若设置成 770，就是只对同组用户开放，也可以写成 ug = rwx，o = 。

4.4.4 文件和文件内容查找

文件查找的命令是

find 路径 -name 需查找的文件名 [-print] [-e 执行命令]

$ find ./ -name *. txt ；在当前路径下查找所有的 . txt 结尾的文件

$ find ./ -name a. txt-exec rm {} \; ；查找当前目录下所有的 a. txt 并删除

$ find ./ -name a. txt-print-exec rm {} \；功能同上并显示目录

$ find ./ -name a. txt-ok rm {} \; ；查找当前目录下所有的 a. txt，经确认删除

文件内容查找命令是：

grep 文件内容 文件名或通配符；文件内容若有空格，需用引号

$ grep“this is a test”a. txt ；查找 a. txt 文件中含有“this is a test”的行

$ grep“this is a test” *. txt ；查找所有 . txt 文件中含有该内容的行

$ ls -l | grep 2009-10 | grep 1298 ；找出当前目录中建立日期为 2009. 10，大小为 1298 的文件。“|”称为管道，就是将上一个程序的结果传给下一个处理

$ ls -l | grep 2009-10 > a. txt ；将当前目录中 2009-10 建立的文件存入 a. txt 文件中。“>”称为转向符，即将屏幕显示的内容转向到文件。

4.4.5 文件压缩与解压缩

1. 单个文件压缩与解压缩

gzip 文件名 ；压缩一个文件

$ gzip a. txt ；压缩文件 a. txt，结果是 a. txt. gz，a. txt 消失

$ gunzip a. txt. gz ；将 a. txt. gz 解压缩，可以不写后缀

gzip -r 目录 ；将一个目录中的每个文件压缩

$ gzip -r a ；将 a 目录中的每个文件压缩

$ gunzip -r a ；将 a 目录中的每个文件解压缩

2. 文件夹压缩

tar [zcvf] 文件名 文件夹 ；将指定文件夹打包压缩成指定文件名

$ tar cvf a. tar a ；将 a 目录打包为 a. tar

$ tar zcvf a. tar. gz a ；将 a 目录打包并压缩为 a. tar. gz

tar [z] tvf 文件名 ；查看压缩打包文件内容

$ tar tvf a. tar ；查看 a. tar 内容

$ tar ztvf a. tar. gz ；查看 a. tar. gz 内容，压缩文件需加 z

tar [zxvf] 文件名 [-C 目录] ；解包在当前目录或指定目录（-C）

$ tar zxvf a. tar. gz ；解压在当前目录

$ tar zxvf a. tar. gz-C ./b ；解压在当前目录的 b 目录下

注意：文件夹打包必须具有对该文件夹的读权限。文件夹解压在某目录，必须具有对该

目录的写的权限。

4.5　项目四完成步骤详解

4.5.1　创建目录和文件

1）创建隶属于财务部门组 CWB 的 账户 CWB-user1，CWB-user2，CWB-user3（具体创建方式见第三讲），切换到某个用户状态下（见图 4-2）。

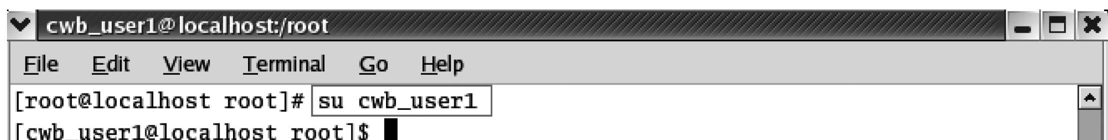


图 4-2　切换用户

2）进入用户主目录/home/cwb_user1，在用户文件夹下创建 Pz，Bb，Tz，Qt 文件夹（见图 4-3）。

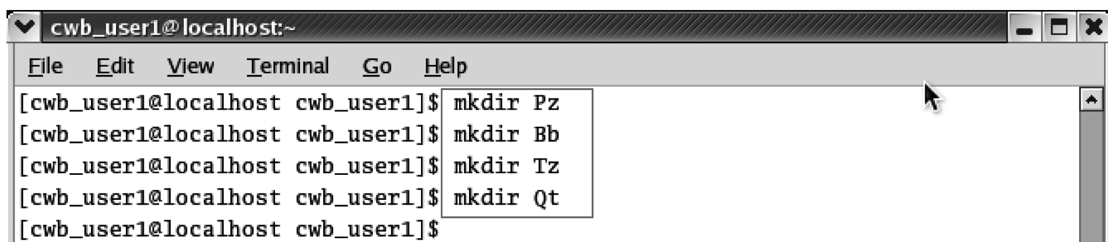


图 4-3　创建目录

创建后查看创建结果（见图 4-4）。

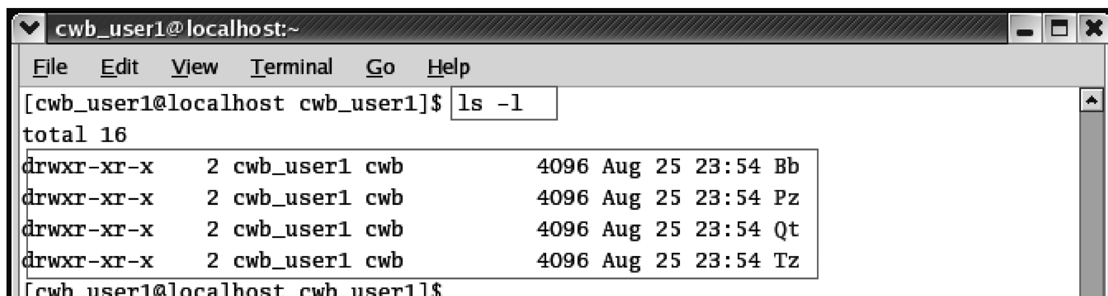


图 4-4　查看文件目录

3）进入到 Pz 文件夹，创建空白文件（见图 4-5）。

查看 Px 文件夹下的文件（见图 4-6）。

4）退出 Pz 文件夹，进入 Bb 文件夹（见图 4-7）。

若要创建有内容的文件，则用 cat 命令创建文件，如下：创建有内容的 Bb1. txt 文件（见图 4-8）。

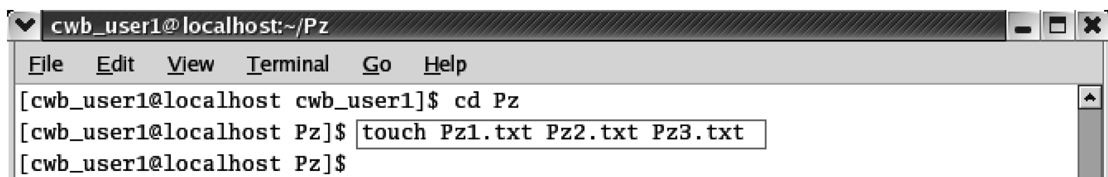


图 4-5 切换目录、创建空文件

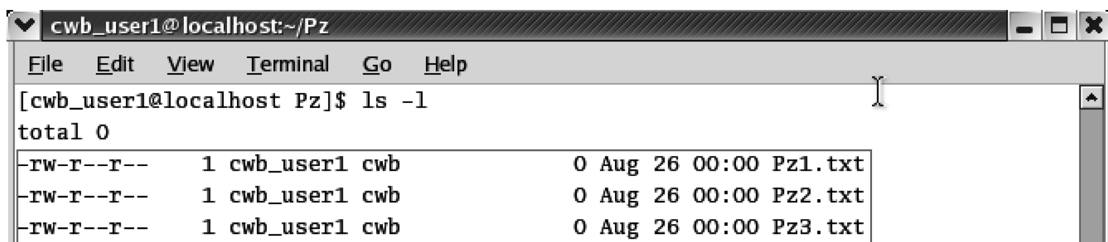


图 4-6 查看创建好的文件

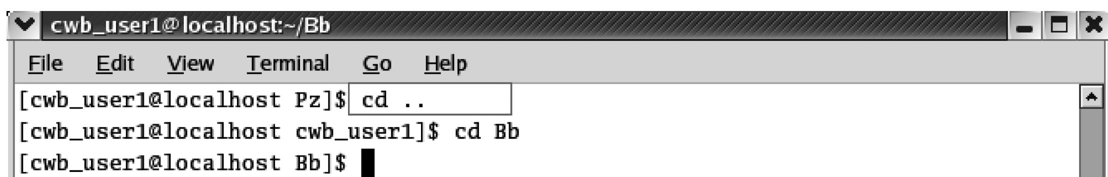


图 4-7 切换至另一目录



图 4-8 用 cat 命令创建文件

输入文件内容结束后按 Ctrl + d 键结束文件内容输入。用同样的方法可以创建其他文件。查看各个文件夹下的文件情况（见图 4-9）。

4.5.2 设置文件/目录权限

因为 Pz，Bb，Tz，Qt 文件夹都在用户主文件夹 cwb_user1 下，该文件夹对除主用户之外的用户皆无任何权限，所以首先应该将该文件夹赋予最大权限，至于每个用户的最终访问权限到后面子文件夹和文件再来设置。设置此文件夹权限如下：

修改文件权限如图 4-10 所示。

从图 4-4 可知默认情况下文件夹对本部门（同组）成员和其他成员可读、不可写，要使得 Pz 目录下的文件在本部门可读、可写且其他无权使用则应该修改文件夹权限。

```
cwb_user1@localhost:~/Qt
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ cd Pz
[cwb_user1@localhost Pz]$ ls -l
total 0
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:00 Pz1.txt
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:00 Pz2.txt    Pz文件夹文件
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:00 Pz3.txt
[cwb_user1@localhost Pz]$ cd ..
[cwb_user1@localhost cwb_user1]$ cd Bb
[cwb_user1@localhost Bb]$ ls -l
total 4
-rw-r--r--    1 cwb_user1 cwb         22 Aug 26 00:11 Bb1.txt
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:13 Bb2.txt    Bb文件夹文件
[cwb_user1@localhost Bb]$ cd ..
[cwb_user1@localhost cwb_user1]$ cd Tz
[cwb_user1@localhost Tz]$ ls -l
total 0
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:14 Tz1.txt
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:14 Tz2.txt    Tz文件夹文件
[cwb_user1@localhost Tz]$ cd ..
[cwb_user1@localhost cwb_user1]$ cd Qt
[cwb_user1@localhost Qt]$ ls -l
total 0
-rw-r--r--    1 cwb_user1 cwb          0 Aug 26 00:14 Qt.txt     Qt文件夹文件
```

图 4-9　查看各目录中的文件

```
root@localhost:~
File  Edit  View  Terminal  Go  Help
[root@localhost root]# chmod 777 /home/cwb_user1
```

图 4-10　修改文件权限

1）修改 Pz 文件夹权限增加同组成员可写，其他成员不可读、不可执行（见图 4-11）。

```
cwb_user1@localhost:~
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ chmod g+w,o-rx Pz
```

```
[zhang@localhost tmp]$ chmod g+w,o-rx Pz
```

图 4-11　修改文件权限

2）修改 Bb 文件夹权限为同组可读、可写，其他可读（见图 4-12）。

```
cwb_user1@localhost:~
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ chmod g=rw,o=r Bb
```

图 4-12　修改文件权限

3）修改 Tz 文件夹权限为同组可读、可写，其他可读（见图 4-13）。



图 4-13 修改文件权限

4）修改 Qt 文件夹权限为同组可读、可写，其他可读、可写（见图 4-14）。

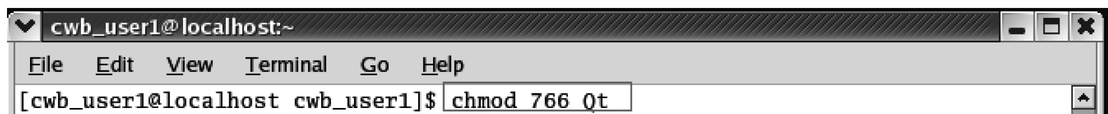


图 4-14 修改文件权限

查看各个文件夹权限（见图 4-15）。

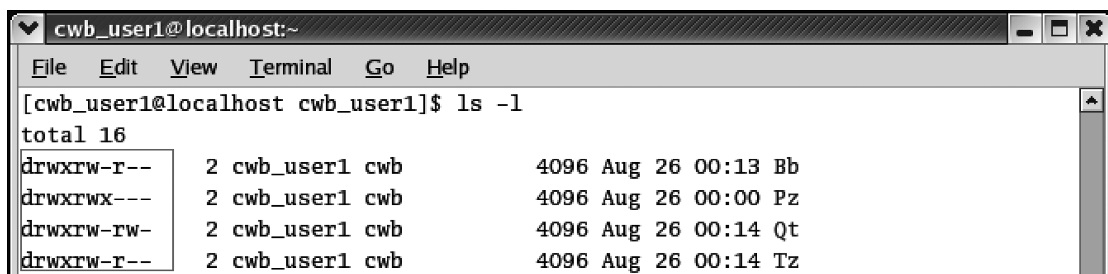


图 4-15 查看文件权限

要想财务部人员不需要复制编辑凭证文件，则需要将 Pz 文件夹进行共享，需要设置该文件夹对同组成员可读、可写，该文件夹下的文件也对同组成员可读、可写。从图 4-9 可知文件在默认情况下是本部门（同组）成员和其他成员只有读权限，所以还需要修改文件权限。

5）修改 Pz 文件夹下 3 个文件为同组可读、可写（见图 4-16）。

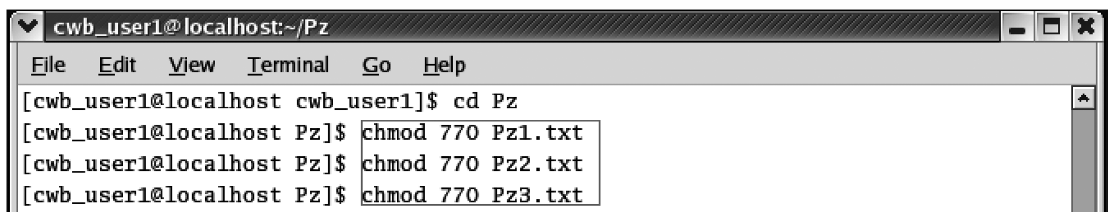


图 4-16 设置文件权限

查看 Pz 文件夹下文件的权限（见图 4-17）。

```
cwb_user1@localhost:~/Pz
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost Pz]$ ls -l
total 0
-rwxrwx---    1 cwb_user1 cwb            0 Aug 26 00:00 Pz1.txt
-rwxrwx---    1 cwb_user1 cwb            0 Aug 26 00:00 Pz2.txt
-rwxrwx---    1 cwb_user1 cwb            0 Aug 26 00:00 Pz3.txt
```

图 4-17　查看文件权限

4.5.3　查找文件和文件内容

如果想要查找当前目录下相关的通知文件，则可以进行如下查找（见图 4-18）。

```
cwb_user1@localhost:~
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ find ./ -name Tz*.txt
./Tz/Tz1.txt
./Tz/Tz2.txt
```

图 4-18　查找文件

若想要查找文件中有“meeting”词汇的文件，可用如下方式查找（见图 4-19）。

```
cwb_user1@localhost:~
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ grep "meeting" Tz/*.txt
Tz/Tz1.txt:all staff should attend meeting on this friday morning in meetingroom
```

图 4-19　查找文件内容

4.5.4　文件压缩与备份

对财务部所有文件夹及下面的文件进行压缩备份文件名为 cwb. tar. gz（见图 4-20）。

```
cwb_user1@localhost:~
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ tar zcvf cwb.tar.gz Pz Bb Qt Tz
Pz/
Pz/Pz1.txt
Pz/Pz2.txt
Pz/Pz3.txt
Bb/
Bb/Bb1.txt
Bb/Bb2.txt
Qt/
Qt/Qt.txt
Tz/
Tz/Tz1.txt
Tz/Tz2.txt
```

图 4-20　用 tar 命令将文件夹压缩到一个文件中

查看备份后的文件（见图 4-21）。

```
cwb_user1@localhost:~
File  Edit  View  Terminal  Go  Help
[cwb_user1@localhost cwb_user1]$ ls -l
total 20
drwxrw-r--    2 cwb_user1 cwb         4096 Aug 26 00:13 Bb
-rw-r--r--    1 cwb_user1 cwb          387 Aug 26 01:27 cwb.tar.gz   备份文件
drwxrwx---    2 cwb_user1 cwb         4096 Aug 26 00:00 Pz
drwxrw-rw-    2 cwb_user1 cwb         4096 Aug 26 00:14 Qt
drwxrw-r--    2 cwb_user1 cwb         4096 Aug 26 00:14 Tz
```

图 4-21　查看压缩结果

若要将备份文件恢复，则采用命令 $ tar zxvf cw. tar. gz。

4.6　思考题

1）采用 root 身份，使用命令 ln-s 建立当前目录下 test（文件属性为 guest）的符号链接 test. link，并将 test. link 的属性由 root 改变为 nobody，则最后文件 test 的属性是什么？

2）在共享文件时，文件夹权限和文件权限是否一定要按照案例方案来设置权限？

3）删除一个非空文件夹该用何种命令？

第5讲

Linux 文件编辑

5.1 引言

5.1.1 Vi 简介

Vi 编辑器是 Visual interface 的简称，通常称之为 Vi。它在 Linux 上的地位就像 Edit 程序在 DOS 上一样。它可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。

Vi 编辑器并不是一个排版程序，它不像 Word 或 WPS 那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。没有菜单，只有命令，且命令繁多。Vi 有 3 种基本工作模式：命令行模式、文本输入模式和末行模式。

Vim 是 Vi 的加强版，比 Vi 更容易使用。Vi 的命令几乎全部都可以在 Vim 上使用。

要在 Linux 下编写文本或语言程序，您首先必须选择一种文本编辑器。可以选择使用 Vim 编辑器的好处是几乎每一个版本的 Linux 都会有它的存在。然而它是在文本模式下使用，需要记忆一些基本的命令操作方式。您也可以选择使用 pico、joe、jove 编辑器，午夜司令官 mc 编辑器等，它们都比 Vim 来得简单。如果您实在不习惯使用文字模式，可以选择视窗环境下的编辑器，像 Gedit、Kate 等，或是使用 KDevelop，它是在 Linux 中的 X Window 下执行的 C/C + + 整合式开发环境。

5.1.2 Vi 使用方法

Vi 编辑器是所有 Unix 及 Linux 系统下标准的编辑器，它的强大不逊色于任何最新的文本编辑器，这里只是简单地介绍它的用法和一小部分指令。由于对 Unix 及 Linux 系统的任何版本，Vi 编辑器是完全相同的，因此可以在其他任何介绍 Vi 的地方进一步了解它。Vi 也是 Linux 中最基本的文本编辑器，学会它后，您将在 Linux 的世界里畅行无阻。

1. Vi 的基本概念

基本上 Vi 可以分为 3 种状态，分别是：

命令模式（command mode）;

插入模式（Insert mode）;

底行模式（last line mode）。

（1）命令模式

控制屏幕光标的移动，字符、字或行的删除，移动复制某区段及进入 Insert mode 下，或者到 last line mode。

（2）插入模式

只有在 Insert mode 下，才可以做文字输入，按 ESC 键可回到命令行模式。

（3）底行模式

将文件保存或退出 Vi，也可以设置编辑环境，如寻找字符串、列出行号等。

不过一般我们在使用时把 Vi 简化成两个模式，就是将底行模式（last line mode）也算入命令行模式（command mode）。

2. Vi 的基本操作

（1）进入 Vi

在系统提示符号输入 Vi 及文件名称后，就进入 Vi 全屏幕编辑画面：

$ Vi myfile

不过有一点要特别注意，就是进入 Vi 之后，是处于命令行模式（command mode），要切换到插入模式（Insert mode）才能够输入文字。初次使用 Vi 的人都会想先用上、下、左、右键移动光标，结果计算机一直哔哔叫，把自己气个半死，所以进入 Vi 后，先不要乱动，转换到插入模式（Insert mode）再说吧！

（2）切换至插入模式（Insert mode）编辑文件

在命令模式（command mode）下按字母 i 键就可以进入插入模式（Insert mode），这时候就可以开始输入文字了。

（3）Insert 的切换

如果目前处于插入模式（Insert mode），就只能一直输入文字，如果发现输错了字，想用光标键往回移动，将该字删除，就要先按 ESC 键转到命令模式（command mode）再删除文字。

（4）退出 Vi 及保存文件

在命令模式（command mode）下，按冒号键进入 Last line mode，例如：

：w filename（输入［w filename］将文章以指定的文件名 filename 保存）

：wq（输入［wq］，存盘并退出 Vi）

：q！（输入 q！，不存盘强制退出 Vi）

3. 命令模式（command mode）**功能键**

（1）插入模式

按切换进入插入模式，按 i 键进入插入模式后是从光标当前位置开始输入文件。

按“a”进入插入模式后，是从目前光标所在位置的下一个位置开始输入文字。

按“o”进入插入模式后，是插入新的一行，从行首开始输入文字。

（2）从插入模式切换为命令行模式

按 ESC 键。

（3）移动光标

Vi 可以直接用键盘上的光标做上、下、左、右移动，但正规的 Vi 是用小写英文字母 h、j、k、l，分别控制光标左、下、上、右移一格。

按［ctrl］+［b］：屏幕往“后”移动一页。

按［ctrl］+［f］：屏幕往“前”移动一页。

按［ctrl］+［u］：屏幕往“后”移动半页。

按［ctrl］+［d］：屏幕往“前”移动半页。

按［gg］：移动到文章的首行（可能只在 Vim 中有效）。

按［G］：移动到文章的最后。

按［ $ ］：移动到光标所在行的“行尾”。

按［^］：移动到光标所在行的“行首”。

按［w］：光标跳到下个字的开头。

按［e］：光标跳到下个字的字尾。

按?：光标回到上个字的开头。

按［#l］：光标移到该行的第#个位置，如 5l，56l。

Vi 还提供了 3 个关于光标在全屏幕上移动并且文件本身不发生滚动的命令。它们分别是 H、M 和 L 命令。

a. H 命令

H 命令将光标移至屏幕首行的行首（即左上角），也就是当前屏幕的第一行，而不是整个文件的第一行。利用此命令可以快速将光标移至屏幕顶部。若在 H 命令之前加上数字 n，则将光标移至第 n 行的行首。值得一提的是，使用命令 dH 将会删除从光标当前所在行至所显示屏幕首行的全部内容。

b. M 命令

M 命令将光标移至屏幕显示文件的中间行的行首。即如果当前屏幕已经充满，则移动到整个屏幕的中间行；如果并未充满，则移动到文本的那些行的中间行。利用此命令可以快速地将光标从屏幕的任意位置移至屏幕显示文件的中间行的行首。例如，在上面屏幕显示的情况下（不论光标在屏幕的何处），在命令模式下，输入命令 M 之后，光标都将移到 add 这一行的字母 a 上。同样值得一提的是，使用命令 dM 将会删除从光标当前所在行至屏幕显示文件的中间行的全部内容。

c. L 命令

当文件显示内容超过一屏时，该命令将光标移至屏幕上的最底行的行首；当文件显示内容不足一屏时，该命令将光标移至文件的最后一行的行首。可见，利用此命令可以快速、准确地将光标移至屏幕底部或文件的最后一行。若在 L 命令之前加上数字 n，则将光标移至从屏幕底部算起第 n 行的行首。同样值得一提的是，使用命令 dL 将会删除从光标当前行至屏幕底行的全部内容。

（4）删除文字

［x］：每按一次，删除光标所在位置的“后面”一个字符。

［#x］：例如，［6x］表示删除光标所在位置的“后面”6 个字符。

［X］：大写的 X，每按一次，删除光标所在位置的“前面”一个字符。

［#X］：例如，［20X］表示删除光标所在位置的“前面”20 个字符。

［dd］：删除光标所在行。

［#dd］：从光标所在行开始删除#行。

（5）复制

［yw］：将光标所在之处到字尾的字符复制到缓冲区中。

［#yw］：复制#个字到缓冲区。

［yy］：复制光标所在行到缓冲区。

［#yy］：例如，［6yy］表示复制从光标所在的该行“往下数”6 行文字。

［p］：将缓冲区内的字符贴到光标所在位置。注意：所有与“y”有关的复制命令都必须与“p”配合才能完成复制与粘贴功能。

（6）替换

［r］：替换光标所在处的字符。

［R］：替换光标所到之处的字符，直到按 ESC 键为止。

（7）回复上一次操作

［u］：如果误执行一个命令，可以马上按下，回到上一个操作。按多次“u”可以执行多次回复。

（8）更改

［cw］：更改光标所在处的字到字尾处。

［c#w］：例如，［c3w］表示更改 3 个字。

（9）撤销更改

［u］：撤销上一次更改，可以一直按“u”，一直撤销到最旧修改状态。

［ctrl］+［r］：恢复撤销的更改，可以一直按该组合键达到最新的改变。

（10）跳至指定的行

［ctrl］+［g］：列出光标所在行的行号。

［#G］：例如，［15G］表示移动光标至文章的第 15 行行首。

4. Last line mode 下命令简介

在使用 last line mode 之前，请记住先按 ESC 键确定已经处于 command mode 下后，再按冒号键即可进入 last line mode。

（1）列出行号

［set nu］：输入“set nu”后，会在文件中的每一行前面列出行号。

（2）跳到文件中的某一行

［#］：［#］号表示一个数字，在冒号后输入一个数字，再按 Enter 键就会跳到该行，如输入数字 15 再按 Enter，就会跳到文章的第 15 行。

（3）查找字符

［/关键字］：先按“/”键，再输入想寻找的字符，如果第一次找的关键字不是想要的，则可以一直按“n”往后寻找直到想要的关键字为止。

［？关键字］：先按“？”键，再输入想寻找的字符，如果第一次找的关键字不是想要的，则可以一直按“n”往前寻找直到想要的关键字为止。

（4）保存文件

［w］：在冒号输入字母“w”就可以将文件保存起来。

（5）离开 Vi

［q］：按“q”就是退出，如果无法离开 Vi，则可以在“q”后跟一个“!”强制离开 Vi。

[wq]：一般建议离开时，搭配“w”一起使用，这样在退出的时候还可以保存文件，可以在后面再跟“!”强制保存退出。

[ZZ]：保存并退出 Vi。

（6）语法加亮

[syntax on]：Vi 编辑器默认不打开语法加亮功能，打开 Vi 编辑器后在 last line mode 下使用 syntax on 命令即可打开语法加亮功能，此时编辑器会高亮显示文件中的关键字，方便编程使用，用 syntax off 命令可关闭该功能。

5.2　企业项目场景 5：Linux 文件编辑

运用 Vi 编辑器编制文本文件，内容如下：

文件名　　order1. txt

文件内容：

Purchasing Order

PO#:1234

Buyer name:Ningbo Haitian import and export Co. ,LTD

Vendor name:Ningbo guanghua spark depot

Item name:Lighter

Color:White

Order quantity:10,000 piece

Delivery:2009/07/01

Ship from:Xiamen

Ship to:Hongkong

Fob:RMB $ 2. 00 not include VAT

Fob point:Xiamen

Total amount:RMB $ 20,000

Payment team:L/C

30% of total amount must be paid before starting to product. 70% of total amount must be paid before shipping

Packing information:

1 piece packed in one small box

box dimenssion:30mm * 15mm * 8mm

5.3　Linux 文件编辑教学目标（4 学时）

1. 教学目标

最终目标：能够用 Vi 编辑器编辑 Linux 系统文件。

促成目标：

1）熟练掌握 Vi 编辑器的使用；
2）熟练掌握多个文件的整合编辑。

2. 工作任务

1）编辑简单文件；
2）修改文件；
3）整合文件。

3. 活动设计

编辑一个实际的文件。

5.4 项目五 实验指导

5.4.1 文件录入命令

i（insert）：在光标前开始输入。
a（append）：在光标后开始输入。
o：从当前行下新的一行开始输入。
O：从当前行上新的一行开始输入。

5.4.2 文件修改

dd：删除光标所在行。
ndd：删除光标所在开始 n 行，n 是一个数字，以下同。
d $：删除光标开始至行未。
nyy：复制从光标开始的 n 行，n 默认就是一行，复制内容在默认缓冲区。
p：将复制或删除内容粘贴到光标所在位置或下一行。
“字母 nyy：将复制内容存储在字母指定的缓冲区。
“字母 p：将字母指定的缓冲区内容粘贴到指定位置。
x：删除光标所在的一个字母。
nx：删除光标开始向后的 n 个字母。
X：删除光标向前的一个字母。
nX：删除光标向前的 n 个字母。
r：修改光标所在的字母，先打 r，再打新的字母，将替换旧字母。
cw：修改光标所在的字，按 Esc 键结束。
ncw：修改光标所在向后的 n 个字，按 Esc 键结束。
c $：修改从光标开始至行最后的内容。
u：撤销刚才的操作，可不断撤销，退回至开始。
。：重复刚才的操作。

5.4.3 文件保存

:x ;存盘退出

:! q	;不存盘退出
:wq	;存盘退出
:w 新文件名	;另存为
:r 文件名	;在光标处插入外部文件内容

5.4.4 文件查找和替换

/查找字符串	;输入/,切换到末行,然后输入需查找的字符串,查找到一个以后,可再按 n 继续向下找,按 Shift + n 键继续向上查找。/可换成?,结果相同,查找方向相反。
:1,$ s/旧字符串/新字符串/g	;1 到最后一行,将所有的旧字符串替换为新字符串,$代表最后

例如:

:1,$ s/student/teacher/g	;将全部 student 替换成 teacher
:5,10s/student/teacher/g	;将 5 ~ 10 行的 student 替换成 teacher
:1,$ s/student/teacher/c	;将全部 student 替换成 teacher,但每个替换需要确认,这里只是把 g 换成 c

5.4.5 其他编辑手段

:set nu	;设置行号,以利于查看,但行号不会存入文件
:行号	;切换到指定行,或采用“行号 G”
0	;将光标移动至行首
$	;将光标移动至行末
b	;光标移动至本字最前面的字母或前一个字
w	;光标移动至下一个字
e	;光标移动至本字最后一个字母下一个字最后
$ vi -o f1 f2 f3	;同时编辑多个文件,用:next	prev 切换

5.5 项目五完成步骤详解

5.5.1 新建文件并打开输入区

如果是在编辑文件前设置好文件名则用如下命令，如图 5-1 所示。

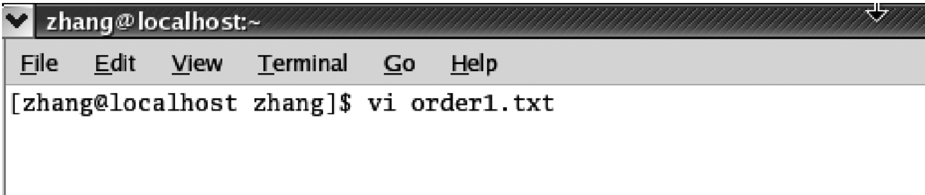


图 5-1 编辑文件命令

若是希望编辑完文件之后保存为一个指定的文件名，则只需要在命令行下输入 Vi 即可。按 Enter 键后打开的工作区如下所示。



此时 Vi 编辑器模式为命令模式，输入“i”，则编辑区左下变成下图样式，如图 5-2 所示。



图 5-2 插入字符

5.5.2 录入文件内容

插入字符和编辑如图 5-3 所示。

5.5.3 保存文件

输入完毕后，按“Esc”键进入命令模式。退出插入状态如图 5-4 所示。

在命令行模式下输入法为半角，输入“：wq”即保存了刚才编辑的文件。若在打开编辑区时没有给文件命名，则在保存时输入“：wq 文件名”即可。

```
zhang@localhost:~
File  Edit  View  Terminal  Go  Help
Purchasing Order
PO# 1234
Buyer name:Ningbo Haitian import and export Co,LTD
Vendor name:Ningbo guanghua spard deport

item:
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
-- INSERT --                                        6,6          All
```

图 5-3　插入字符和编辑

```
zhang@localhost:~
File  Edit  View  Terminal  Go  Help
Purchasing Order
PO# 1234
Buyer name:Ningbo Haitian import and export Co,LTD
Vendor name:Ningbo guanghua spard deport

item:
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
                                                    6,5          All
```

图 5-4　退出插入状态

5.6 思考题

1）如何用 Vi 编辑器同时打开多个文件？

2）在 Vi 编辑器中如何进行多个打开的文件切换？

3）在 Vi 编辑器中如何进行查找和替换文字？

第6讲

Linux 系统管理

 导读

Linux 系统管理指对 Linux 操作系统的环境进行管理，以保证系统的正常运行。主要内容包括磁盘配额管理、定时任务执行、系统备份和还原等。

6.1 引言

如果 Linux 服务器有多个用户经常访问数据时，为了维护所有用户对硬盘空间的公平使用，磁盘配额（Quota）就是一项非常有用的工具。在 Linux 系统中，由于是多用户、多任务环境，所以会有多用户共同使用一个硬盘空间的情况发生，如果其中有少数几个用户大量占用了硬盘空间，就会影响其他用户的使用权限。因此，管理员应该适当限制用户的硬盘空间使用，以妥善分配系统资源。

我们可能会发现，为什么系统经常会主动执行一些任务？这些任务是如何设置的和如何工作的？如果想要在某一时刻进行系统的自动备份工作而不需要亲自动手启动它，该如何处理呢？这些例行工作可分为“一次性”的和“循环”的工作这是由系统的哪些服务负责完成的？

Linux 例行性工作进行的调度就是将这些工作安排执行的流程，Linux 的调度就是通过 crontab 和 at 来实现的。at 是个可以处理仅执行一次就结束调度的命令。crontab 这个命令所设置的工作会一直进行下去，可循环的时间分为分钟、小时、每周、每月、每年等。

6.1.1 什么是磁盘配额（Quota）

Quota 就字面上的意思来看，就是“限额”的意思。如果是用在零用钱上面，就是“有多少零用钱一个月”的意思。如果是在计算机主机上的磁盘使用量上，以 Linux 来说，就是有多少容量限制的意思。可以使用 Quota 来让磁盘的容量使用较为公平。

举例来说，用户的默认主文件夹是在/home 下面，如果/home 是个独立的分区，容量大小为 10GB，而在/home 下有 20 个用户，每个用户平均就应该有 500MB 的空间，若有一个不讲理的用户，独自在文件夹下面下载了很多电影，占了 9GB 的空间，这样一来，其他用户的可用空间就少得可怜了。如果要使磁盘的容量公平分配，就要靠

Quota 来帮助解决了。

6.1.2 Quota 的一般用途

Quota 比较常用的几种情况是：

1）针对 Web 服务器。例如，每个人的网站空间容量限制。

2）针对 Mail 服务器。例如，每个人的邮件空间容量限制。

3）针对文件服务器。例如，每个人最大的可用网络硬盘空间。

以上是对网络服务的设计，如果是针对 Linux 系统主机上的设置，则用途有下面这些：

1）限制某一用户组所能使用的最大磁盘空间配额（使用用户组限制）。可以将主机上的用户分门别类，比较喜好的那一群的使用配额就可以高一些。

2）限制某一用户的最大磁盘配额（使用用户限制）。在限制了用户组之后，可以继续对个人用户进行限制，使得同一组下的用户有不同的限制。

6.1.3 Quota 的使用限制

虽然 Quota 很好用，但是使用上还是有些限制需要事先了解：

1）仅针对整个文件系统；

2）内核必须支持 Quota；

3）只对一般身份用户有效，并不是所有的 Linux 上的账户都可以设置 Quota，例如，root 用户就不能设置 Quota，因为整个系统所有的数据几乎都是它的；

4）不能对某个目录设置 Quota，但可以针对某个文件系统来设置，这就是为什么目录与挂载点还有文件系统有关系。

6.1.4 Quota 的规范设置选项

Quota 针对整个文件系统的限制主要分为下面几部分：

（1）容量限制或文件数量限制（block 或 inode）

文件系统主要规划为存放属性 inode 与实际文件数据 block 块，Quota 既然是管理文件系统，当然也可以管理 inode 和 block。这两个管理的功能为

1）限制 inode 用量：管理用户可以新建的文件数量；

2）限制 block 用量：管理用户磁盘容量的限制，这种方式较为常见。

（2）软限制和硬限制（soft/hard）

既然是规范，当然就有限制值。不管是 inode/block，限制值都有两个，分别是 soft 与 hard。通常 hard 限制值要比 soft 还要高。举例说，若限制项目为 inode，可以限制 hard 为 100，而 soft 为 80。这两个限值的意义如下：

1）hard：表示用户的绝对用量不会超过这个限制值，以上面的设置为例，文件数不能超过 100 个，若超过这个值则系统就会锁定该用户的磁盘使用权；

2）soft：表示用户在低于 soft 限制值时，可以正常使用磁盘，但若超过 soft 值且低于 hard 限值时（介于 80 ~ 100），每次用户登录系统时，系统会主动发出磁盘即将爆满的警告信息，且会给一个宽限时间（gracetime）。不过，若用户在宽限时间倒数期间就将容量再次

降低至 soft 限值之下，则宽限时间就会停止。

（3）会倒计时的宽限时间（gracetime）

1）上面说到的宽限时间，是只有用户在磁盘用量介于 soft 到 hard 之间时，才会出现的会倒计时的一个宽限时间。由于到了 hard 限值时，用户的磁盘使用权就会被锁住。为了防止用户没有注意到磁盘配额的问题出现，因此设计了 soft。当磁盘用量即将到达 hard 且超过了 soft 时，系统就会给予警告，但也会给一段时间让用户自行管理磁盘。一般默认的宽限时间为 7 天，如果 7 天内都不进行任何磁盘管理，那么 soft 限制就会替代 hard 限制作为 Quota 的限制。

2）以上面的例子为例，假如文件数已经达到 90 个，超过了 soft 的值，但仍低于 hard 的值 100，那么 7 天的宽限时间就会开始倒计时，若 7 天内没有删除任何文件来为文件系统瘦身，那么 7 天后文件用量最大值就变成 80（就是 soft 值），磁盘使用权就会被锁住而无法新增文件了。

6.1.5　Linux 工作调度种类：at，crontab

我们可以清楚地发现一般有两种工作调度方式：

1）一种是例行性的，就是每隔一定的周期要来办理的事项；

2）一种是突发性的，就是做完这次以后就没有了那种。

在 Linux 系统中如何实现这两个功能呢？那就是使用 at 和 crontab 这两个命令来实现。

at：可以处理仅执行一次就结束的命令。

crontab：所设置的工作将会循环执行下去。可循环的时间分为分钟、小时、每周、每月或每年等。

6.1.6　at 运行方式

既然是工作调度，就应该会有工作的方式，并且将这些工作排进工作日程表中。那么生成工作的方式是怎么进行的呢？事实上，我们使用 at 这个命令来生成所需要的工作，并将这个工作以文本文件的方式写入/var/spool/at/目录内，该工作便等待 at 这个服务的取用与执行，就这么简单。

不过不是所有的人都可以进行 at 工作调度，因为安全的需要，很多主机被所谓“绑架”后，最常发现的就是系统中多了很多黑客程序，这些程序非常可能运用工作调度来执行或收集系统信息，并定时报给黑客团体。

怎样才能安全地使用 at 管理呢？可以利用/etc/at. allow 与/etc/at. deny 这两个文件来进行 at 的使用限制。加上这两个文件后，at 的工作情况就是这样的：

1）先寻找/etc/at. allow 这个文件，写在这个文件中的用户才能使用 at，没有在这个文件中的用户则不能使用 at（即使没有写在/etc/at. deny 文件中）。

2）如果/etc/at. allow 不存在，就寻找/etc/at. deny 这个文件，写在/etc/at. deny 文件中的用户不能使用 at，而没有写则这个文件中的用户就可以使用 at。

3）如果以上两个文件都不存在，那么只有 root 可以使用 at 命令。

从以上规则可以看出，/etc/at. allow 的管理比较严格，而/etc/at. deny 的管理比较宽松（账号不在文件中就能使用 at 命令）。在一般的套件中，由于假设系统上的所有用户都

是可信任的，因此系统通常会保留一个空的/etc/at. deny 文件，这样就允许所有的用户使用 at 命令。如果不希望某个用户使用 at 命令，就将该用户账号写入/etc/at. deny，一个账号写一行。

当使用 at 时会进入一个 Shell 环境。建议最好使用绝对路径来执行命令，这样问题会比较少。由于命令的执行与 PATH 变量有关，同时与当时的工作目录有关，因此，使用绝对路径执行命令是比较可靠的方法。举例来说，在/home/user1/下执行“at 12：00”然后输入“mail user2 < test. txt”，请问，那个 test. txt 文件会在哪里呢？答案是在“/home/user1/test. txt”。这是因为 at 在运行时会到当时执行 at 命令的那个工作目录的缘故。

有些用户希望在某些时候在终端屏幕上显示“welcome”的字样，然后就在 at 命令中执行“echo welcome”这样的命令，可是时间到了，却发现没有任何信息在屏幕上出现，这是为什么呢？这是因为 at 的执行与终端机环境无关，而所有的标准输出与标准错误输出都将传送到执行者的 mailbox 中去。所以在终端机上当然看不到任何信息。那要如何处理呢？可以通过终端机的设备来处理。例如在 tty1 登录，就可以使用“echo welcome > /dev/tty1”来输出信息。

6. 1. 7 cron 运行方式

相对于 at 仅执行一次的工作，循环执行的例行工作调度则是由 cron（crond）这个系统服务来控制的。Linux 系统中原本就有非常多的例行性工作，因此这个系统服务是默认启动的。由于用户也可以自己进行例行性工作调度，系统提供给用户例行性工作调度的命令是 crontab。

不过，为了安全起见，与 at 命令类似，可以限制使用 crontab 的用户账号。使用的限制文件是：

● /etc/cron. allow

■ 将可以使用 crontab 的账号写入其中，若不在这个文件中的用户则不得使用 crontab。

● /etc/cron. deny

■ 将不可以使用 crontab 的账号写入其中，若未记录在该文件中的用户，就可以使用 crontab。

同样，以优先级来说，/etc/cron. allow 比/etc/cron. deny 要优先。从判断上说，这两个文件只用了一个而已。因此，建议保留其中一个，以免影响自己在设置上的判断。一般来说，系统默认保留/etc/cron. deny 文件，可以将不让执行 crontab 的用户写入/etc/cron. deny 中，一个账号一行。

当用户使用 crontab 这个命令建立新的工作调度后，该项工作就会被记录到/var/spool/cron/目录中，而且是以账号作为判别的。举例来说，user1 使用 crontab 后，他的工作会被记录到/var/spool/cron/user1 中。但请注意，不要使用 Vi 直接编辑该文件，以免引入语法错误，导致系统无法执行 cron。另外，cron 执行的每项工作都会被记录到/var/log/cron 这个日志文件中，所以，想知道 Linux 系统是否被植入木马，也可以查询/var/log/cron 日志文件。

6.2　企业项目场景 6：Linux 系统管理

Linux 系统安装后，为了保证系统的安全和合理的使用，需要为每个部门和用户分配磁盘使用空间，部门及用户空间分配见表 6-1。

表 6-1　部门及用户空间分配

部 门 名 称	部门可用空间/MB	用户可用空间/MB	备　注
总经理办公室	50	15	3 用户
财务部	200	60	3 用户
后勤部	60	20	3 用户
业务 1 部	100	20	5 用户
业务 2 部	100	20	5 用户

其他要求：

1）定时执行目录清理任务（例如：对 Qt 目录定时清理）；

2）定时执行系统目录备份任务（例如：对“财务凭证”、“财务报表”目录定时备份）；

3）若遇系统故障，需要对备份文件进行恢复。

6.3　Linux 系统管理教学目标（8 学时）

1. 教学目标

最终目标：能对磁盘配额进行设置，定时执行命令，系统备份和还原。

促成目标：

1）熟悉磁盘配额方法；

2）熟悉定时执行命令方法；

3）熟悉系统备份和还原方法。

2. 工作任务

1）进行磁盘配额（按组或用户）；

2）设置定时执行命令；

3）进行系统备份和还原。

3. 活动设计

1）设置磁盘备份；

2）设置定时系统备份。

6.4 项目六　实验指导

6.4.1　磁盘配额

（1）配置/etc/fstab 文件

```
#vi /etc/fstab              ;编辑/etc/fstab 文件,将/分区文件系统配置改为
原来:
LABEL=/     /     ext3     defaults                              1     1
改为:
LABEL=/     /     ext3     defaults, usrquota, grpquota          1     1
改完后重新挂接/文件系统:
#mount  -o remount  /          ;重新挂接根文件系统
```

（2）进行配额检查，生成配额文件

```
#quotacheck  -avum            ;生成用户配额文件 aquota. user
#quotacheck  -avgm            ;生成组配额文件  aquota. group
完成后重启系统
#init 6                       ;重启系统,使配置生效
```

（3）新建测试组和用户

```
#groupadd tgrp                ;新建测试组 tgrp
#useradd  -g tgrp user1       ;新建用户 user1,并指定组为 tgrp
#useradd  -g tgrp user2
```

（4）配置用户的磁盘配额

```
#edquota [-u] user1           ;配置用户 user1 的磁盘配额,在编辑文件中分别设置
                               磁盘容量和文件数,blocks 代表允许磁盘容量,inodes
                               代表允许的文件数,soft 代表软极限,hard 代表硬极
                               限,如果没有延展期,软极限和硬极限是一样的。延
                               展期用 edquot-t(user)或 edquota-tg(group)来设置。
```

（5）配置组的磁盘配额

```
#edquota  -g 组名
#edquota  -g tgrp             ;配置组 tgrp 的磁盘配额,要求同用户配置。
这里 1000blocks = 1MB
```

（6）显示磁盘配额

```
#quota  -u 用户名             ;显示用户磁盘配置情况
#quota  -g 组名               ;显示组磁盘配置情况
```

6.4.2　定时执行命令

（1）定时执行临时命令

用 at 命令可以设置一个临时任务在指定时间执行。设置方法如下：

```
#at  时间                     ;设置执行时间
```

at > 执行任务 ;设置执行命令
at > Ctrl + D ;结束
例如：
#at 12:00 10/1/2009 ;时间 2009. 10. 1 12 时
at > init 6 ;设置系统重启
at > Ctrl + D ;结束
（2）时间设置方法
at now + 5 minutes ;5 分钟后执行
at now + 1 hour ;1 小时后执行
at now + 3 days ;3 天后执行
at 15:30 ;当天下午 3 点 30 分执行
at 1:00 10/25/2009 ;在 2009. 10. 25 1 点执行
（3）任务查看
#atq ;查看任务，显示如下
10 2009-10-2512:30 a root ;a 代表 at 任务，b 代表 batch（空闲执行）任务
（4）任务删除
#atrm 任务号
#atrm 10 ;删除 10 号任务

6.4.3 周期执行命令

（1）用户周期性任务
$ crontab -e ;编辑用户周期性任务文件
文件格式：

分	小时	天	月	星期	命令或命令文件
0-59	0-23	1-31	1-12	0-7	

通配符 * 代表每分钟、小时、天、月、星期，例如：
* * * * * date > > date. txt ;每分钟读取日期时间加入文件 data. txt 中。
* * */2 * * tar zcvf home. tgz /home ;每隔一天做一次/home 目录的备份。
（2）周期性任务管理
所有的任务都会放在/etc/spool/cron 目录下。
#crontab -l ;查看用户设置的周期性任务，普通用户也一样
#crontab -r ;删除周期性任务设置

6.5 项目六完成步骤详解

6.5.1 为部门和用户分配磁盘空间

1）配置系统文件/etc/fstab（见图 6-1）。

```
文件(F)  编辑(E)  查看(V)  终端(T)  标签(B)  帮助(H)
# This file is edited by fstab-sync - see 'man fstab-sync' for details
LABEL=/              /                  ext3    defaults,usrquota,grpquota      1 1
none                 /dev/pts           devpts  gid=5,mode=620  0 0
none                 /dev/shm           tmpfs   defaults        0 0
none                 /proc              proc    defaults        0 0
none                 /sys               sysfs   defaults        0 0
LABEL=SWAP-sda2      swap               swap    defaults        0 0
/dev/hdc             /media/cdrom       auto    pamconsole,fscontext=system_u:object_r:r
emovable_t,exec,noauto,managed 0 0
/dev/fd0             /media/floppy      auto    pamconsole,fscontext=system_u:object_r:r
emovable_t,exec,noauto,managed 0 0
~
```

图 6-1　配置磁盘分配系统文件

2）重新挂接文件系统和配额检查（见图 6-2）。

```
[root@fanhs ~]# mount -o remount /
[root@fanhs ~]# quotacheck -avum
quotacheck: Scanning /dev/sda1 [/] quotacheck: Cannot stat old user quota file:
录
quotacheck: Cannot stat old group quota file: 没有那个文件或目录
done
quotacheck: Checked 16932 directories and 137969 files
quotacheck: Old file not found.
[root@fanhs ~]# quotacheck -avgm
quotacheck: Scanning /dev/sda1 [/] quotacheck: Cannot stat old group quota file
目录
done
quotacheck: Checked 16932 directories and 137970 files
quotacheck: Old file not found.
[root@fanhs ~]# 
```

图 6-2　重新挂接文件系统和配额检查

完成后重启系统，使系统配置生效。

3）新建部门（组）和用户（见图 6-3）。

总经理办公室	zjl	用户	zjl1,zjl2,zjl3
财务部	cwb	用户	cwb1,cwb2,cwb3
后勤部	hqb	用户	hqb1,hqb2,hqb3
业务 1 部	yw1b	用户	yw1b1,yw1b2,yw1b3
业务 2 部	yw2b	用户	yw2b1,yw2b2,yw2b3

```
[root@fanhs ~]# groupadd zjl
[root@fanhs ~]# groupadd cwb
[root@fanhs ~]# groupadd hqb
[root@fanhs ~]# groupadd yw1b
[root@fanhs ~]# groupadd yw2b
[root@fanhs ~]# useradd -g zjl zjl1
[root@fanhs ~]# useradd -g zjl zjl2
[root@fanhs ~]# useradd -g zjl zjl3
[root@fanhs ~]# useradd -g cwb cwb1
[root@fanhs ~]# useradd -g cwb cwb2
[root@fanhs ~]# useradd -g cwb cwb3
[root@fanhs ~]# useradd -g hqb hqb1
[root@fanhs ~]# useradd -g hqb hqb2
[root@fanhs ~]# useradd -g hqb hqb3
[root@fanhs ~]# useradd -g yw1b yw1b1
[root@fanhs ~]# useradd -g yw1b yw1b2
[root@fanhs ~]# useradd -g yw1b yw1b3
[root@fanhs ~]# useradd -g yw2b yw2b1
[root@fanhs ~]# useradd -g yw2b yw2b2
[root@fanhs ~]# useradd -g yw2b yw2b3
```

图 6-3　新建组和用户

4）为各部门分配磁盘空间。查看组空间使用情况如图 6-4 所示。

部门名称	部门可用空间	用户可用空间
总经理办公室	50M	15M_
财务部	200M	60M_
后勤部	60M	20M_
业务 1 部	100M	20M_
业务 2 部	100M	20M_

总经理办公室：

#edquota　-g zjl

```
Disk quotas for group zjl (gid 502):
  Filesystem                   blocks       soft       hard     inodes     soft
    hard
  /dev/sda1                       276      50000      50000         33        0
       0
~
```

图 6-4　查看组空间使用情况

查看用户空间使用情况如图 6-5 所示。

#edquota -u zjl1

```
Disk quotas for user zjl1 (uid 502):
  Filesystem                   blocks       soft       hard     inodes     soft
    hard
  /dev/sda1                        96      15000      15000         12        0
       0
```

图 6-5 查看用户空间使用情况

其他用户设置相同。

查看财务部组空间使用情况如图 6-6 所示。

#edquota -g cwb

```
Disk quotas for group cwb (gid 503):
  Filesystem                   blocks       soft       hard     inodes     soft
    hard
  /dev/sda1                      276     200000     200000         33
        0
~
```

图 6-6 查看财务部组空间使用情况

查看财务部用户空间使用情况如图 6-7 所示。

#edquota -u cwb1

```
Disk quotas for user cwb1 (uid 505):
  Filesystem                   blocks       soft       hard     inodes     soft
    hard
  /dev/sda1                        96      60000      60000         12        0
       0
```

图 6-7 查看财务部用户空间使用情况

其他部门和用户设置相同。

6.5.2 定时执行任务

（1）定时执行任务设置

任务 1：在 2010 年 12 月 31 日 12 时删除/home/cwb1/qt 目录中所有内容。

设置定时任务如图 6-8 所示。

```
[root@fanhs ~]# at 12:00 12/31/2010
at> rm -r /home/cwb1/qt/*
at> <EOT>
job 3 at 2010-12-31 12:00
[root@fanhs ~]#
```

图 6-8 设置定时任务

任务 2：8 小时后备份财务部报表目录/home/cwb1/cwbb，文件名为 cwbb. tgz。

设置定时任务如图 6-9 所示。

```
[root@fanhs ~]# at now + 8 hour
at> tar zcvf cwbb.tgz /home/cwb1/cwbb
at> <EOT>
job 5 at 2010-08-31 17:33
[root@fanhs ~]#
```

图 6-9　设置定时任务

（2）周期性执行任务设置

任务 1：每周一次（每隔 7 天）早上 8 时清理财务部 qt 目录/home/cwb1/qt，删除所有内容。

周期性定时任务如图 6-10 所示。

#crontab　-e

```
0       8       */7     *       *       rm -r /home/cwb1/qt
~
```

图 6-10　周期性定时任务

任务 2：每月 30 日下午 5 时做财务部目录/home/cwb1 备份，备份文件名为 cwb. tgz。

周期性定时任务如图 6-11 所示。

```
*       *       */7     *       *       rm -r /home/cwb1/qt
0       17      30      *       *       tar zcvf cwb.tgz /home/cwb1
```

图 6-11　周期性定时任务

注意：周期性任务可以有多个。

任务 3：每月 1 日早上 8 时，恢复财务部报表目录/home/cwb1/cwbb，恢复文件名及路径在/root/cwbb. tgz。

周期性定时任务如图 6-12 所示。

```
*       *       */7     *       *       rm -r /home/cwb1/qt
0       17      30      *       *       tar zcvf cwb.tgz /home/cwb1
0       8       1       *       *       tar zxvf /root/cwbb.tgz -C /home/cwb1/cwbb
~
```

图 6-12　周期性定时任务

6.6 思考题

1）如何设置用户和组的磁盘配额？

2）定时执行命令有几种方法？各有什么特点？

3）若周期性地进行备份，应如何设置？

第7讲 Linux 软件包安装

 导读

Linux 系统软件包安装与管理与 Windows 系统有很大区别，安装的方法也很特殊。本章学习在 Linux 系统环境下如何安装、删除、升级、验证软件包，主要学习 RPM 方式的软件包安装管理。

7.1 引言

在 Windows 下安装软件时，只需运行软件的安装程序（Setup、Install 等）或者用解压缩软件解开即可安装，运行反安装程序（Uninstall、Unware、卸载等）就能将软件清除干净。这些完全图形化的操作接口，简单到只要用鼠标一直单击【下一步】按钮就可以了。而 Linux 就不一样了，很多的初学者都抱怨在 Linux 下安装和卸载软件非常困难，没有像使用 Windows 时那么直观。

在 Linux 系统中，软件安装程序比较复杂，不过最常见的有两种：一种是软件的源代码，您需要自己动手编译它。这种软件安装包通常是用 gzip 压缩过的 tar 包（后缀名为 . tar. gz）。另一种是软件的可执行程序，你只要安装它就可以了。这种软件安装包通常是一个 RPM 包（RedHat Linux Packet Manager，就是 RedHat 的软件包管理器），后缀名是 . rpm。当然，也有用 RPM 格式打包的源代码、用 gzip 压缩过的可执行套装程序。只要理解了以下的思路，这两种形式的安装包也就不在话下了。

7.1.1 RPM 软件包管理

RPM 是 RedHat 公司开发的软件包管理器，使用它可以很容易地对 RPM 形式的软件包进行安装、升级、卸载、验证、查询等操作，安装简单，而卸载时也可以将软件安装在多处目录中的文件删除干净，因此推荐初学者尽可能使用 RPM 形式的软件包。软件包往往有特定的命令规范，名字是由“文件名 + 版本号 + . rpm”组成的字串，例如 apache-3. 1. 12-i386. rpm 和 apache-devel-3. 1. 12-i386. rpm，它们的软件包名称分别是 apache 和 apache-devel。各个支持 RPM 格式的 Linux 常见软件的 RPM 包可以在网站 rpmfind. net 中找到。

RPM 全名是“RedHat Package Manager”，简称 RPM。顾名思义，当初这个软件管理的机制是由 Red Hat 这家公司发展而来的。RPM 是以一种数据库记录的方式将需要的软件安装

到 Linux 系统的一套管理机制。

RPM 最大的特点就是将要安装的软件事先编译，并且打包成为 RPM 机制的安装包，通过包装好的软件中默认的数据库记录这个软件要安装的时候必须具备的依赖属性软件，当安装在 Linux 主机时，RPM 会先依照软件里的数据查询 Linux 主机的依赖属性是否满足，若满足则予以安装，若不满足则不予安装。安装的时候就将该软件的信息整个写入 RPM 的数据库中，以便未来的查询、验证与反安装。这样的优点是：

1）由于已经编译完成并且打包完毕，所以软件传输与安装上很方便（不需要重新编译）。

2）由于软件的信息已经记录在 Linux 主机的数据库中，很方便查询、升级与反安装。

但是这样也造成许多困扰。由于 RPM 文件是已包装好的数据，也就是说，里面的数据已经“编译完成”了，所以该软件几乎只能在原本默认的硬件与操作系统版本中使用。也就是说，主机系统环境必须要与当初这个软件文件的主机环境相同才行，所以通常不同的套件所发布的 RPM 文件并不能用在其他的套件上。更有甚者，相同的套件不同的版本之间也无法互通。因此，可以发现这些软件管理机制的问题是：

1）软件安装的环境必须与打包时的环境需求一致和相当；

2）需要满足软件的依赖属性需求；

3）反安装时需要特别小心，最底层的软件不可先删除，否则可能造成整个系统的问题。

为了解决软件之间具有相关性的问题（就是所谓软件的依赖属性），PRM 就在提供打包软件时，同时加入一些信息记录功能，包括软件的版本、打包软件者、依赖属性、功能说明、本软件的所有文件记录等，然后在 Linux 系统上面也建立一个 RPM 软件数据库，如此一来，当要安装某个以 RPM 类型提供的软件时，在安装过程中 RPM 会去检验数据库里面是否已经存在相关的软件，如果数据库显示不存在，那么这个 RPM 文件“默认”就不能安装，这个就是 RPM 类型的文件的“软件的属性依赖”问题。

为了重复利用既有的软件功能，很多软件都会以函数库的方式释出部分功能，以方便其他软件的调用，例如 PAM 模块验证功能。此外，为了节省用户的数据量，目前的套件在发布软件时，都会将软件的内容分为一般使用和开发使用两大类。所以才会经常看到有类似 Pam-x. x. rpm 与 Pam-devel-x. x. rpm 之类的文件名，而默认情况下，大部分的 Software-devel-x. x. rpm 都不会安装，因为终端用户大部分不会去开发软件。

因为有了上述现象，因此，RPM 软件文件就会有所谓属性依赖的问题产生（其实所有的软件管理几乎都有这方面的情况存在）。那有没有解决办法呢？前面不是谈到 RPM 软件文件内部会记录依赖属性的数据吗？要是将这些依赖属性的软件首先列表，在有需要安装软件的时候先到这个列表去找，同时与系统内部已安装的软件相比较，没有安装的软件就同时安装起来，那不就解决了依赖属性问题了吗？有没有这种机制呢？答案是有的，那就是 YUM 机制。

首先将发布的软件放置到 YUM 服务器内，然后分析这些软件的依赖属性问题，将软件内的记录信息写下来，再将这些信息分析后记录成软件相关性清单列表。这些列表数据与软件所在的位置可以称为容器。当客户端有软件安装的需求时，客户端主机会主动从网络上的 YUM 服务器的容器网址下载清单列表，通过该列表数据与本机 RPM 数据库已存在的软件数据相对比，就能够安装所有需要的具有依赖属性的软件。

当客户端有升级、安装的需求时，YUM 会向容器要求清单的更新，等到清单到达本机后，与本机的 RPM 数据库进行比较，就知道该下载什么软件了。接下来 YUM 会跑到容器服务器（YUM SERVER）下载所需要的软件，然后通过 RPM 机制开始安装软件，这就是整个 YUM 在线升级的流程。

7.1.2　Tarball 源代码软件包

Linux 软件的源代码分发是指提供了该软件所有程序源代码的发布形式，需要用户自己编译成可执行的二进制码并进行安装。其优点是配置灵活，可以随意去掉或保留某些功能/模块，适应多种硬件/操作系统平台及编译环境；缺点是难度较大，一般不适合初学者使用。

以 Tarball 方式释出的软件是需要重新编译成可执行的二进制文件的。而 Tarball 是以 tar 命令打包与压缩的文件，所以，就需要先将 Tarball 解压缩，然后到源码所在的目录进行 makefile 的建立，再以 make 来进行编译和安装操作。整个安装的基础操作是这样的：

1）取得源文件，将 tarball 文件在/usr/local/src 目录下解压。

2）进入新建的目录下面，查询 install 与 readme 等相关文件内容（很重要的步骤）。

3）相关属性软件的安装，根据 install 或 readme 的内容查找并安装好相关的软件（非必要）。

4）建立 makefile，以自动检测程序（configure 和 config），检测操作环境，并建立 Makefile 文件。

5）编译，以 make 这个程序并使用该目录下的 Makefile 作为它的参数配置文件来进行 make（编译或其他）的操作。

6）安装，以 make 这个程序，并以 Makefile 参数配置文件，依据 install 这个目标指定安装到正确的路径。

注意上面的第 2 个步骤，通常在每个软件释出的时候，都会附上 install 或 readme 这种文件名的帮助文件，请详细阅读一遍，通常这些文件会记录这个软件的安装需求、软件的工作项目与安装的参数设置及技巧等，只要仔细读完这些文件，基本上要安装好 tarball 的文件都不会有什么大问题。

Makefile 制作出来后，里面会有相当多的目标，最常见的是 install 与 clean。通常 makeclean 代表将目标文件（object file）清除掉，make 则是将源代码进行编译而已。注意，编译完成的可执行文件与相关的配置文件还在源代码所在的目录中。因此，最后进行“makeinstall”来将编译完成的所有内容都安装到正确路径，这样就可以使用该软件了。

下面简要介绍大部分 tarball 软件安装的命令执行方式：

（1）./configure

这个步骤就是建立 Makefile 文件。通常程序开发者会写 script 来检查 Linux 系统、相关的软件属性等，这个步骤相当重要，因为将来的安装信息都是在这一步骤内完成的。另外，这个步骤的相关信息应该参考该目录下的 readme 或 install 相关的文件。

（2）makeclean

make 会读取 Makefile 中关于 clean 的工作。这个步骤不一定会有，但是希望执行一下，因为它可以去除目标文件。因为谁也不确定源码里面到底有没有包含上次编译过的目录文件（*.o），所以还是清除比较妥当。至少等新编译出来的执行文件可以确定是使用自己的机器所编译完成的。

（3） make

make 会依据 Makefile 当中的默认工作进行编译。编译的工作主要是 gcc 将源代码编译成为可以被执行的目标文件，但是这些目标文件通常还需要一些函数库之类的链接后，才能生成一个完整的可执行文件。使用 make 就是要将源码编译成为可执行文件，而这个可执行文件会放置在目前所在目录之下，尚未被安装到预定安装目录中。

（4） make install

通常这就是最后的安装步骤，make 会依据 Makefile 文件中关于 install 的选项，将上一个步骤所编译完成的数据安装到默认的目录中，这就完成了安装。

请注意，上面的步骤是一步一步来进行的，而其中只要一个步骤无法完成，那么后续的步骤就完全没有办法进行。因此，要确定每一步骤都是成功的才可以。举例来说，如果在 ./configure 没有成功，那么就表示 Makefile 无法被建立起来，要知道，后面的步骤都是根据 Makefile 来进行的，既然 Makefile 无法建立，后续的步骤也无法进行。

另外，如果在 make 无法成功，就表示文件无法被编译成可执行文件，那么 make install 主要是将编译完成的文件放置到文件系统中。既然都没有可用的执行文件，怎么进行安装？所以，要每一个步骤都正确无误才能往下继续进行。

7.2　企业项目场景 7：Linux 系统软件包安装

为了适应系统功能的增加和更新，需要管理 Linux 系统的软件包，对部分软件包进行安装、删除、升级操作，并检查软件包的现状，以保证系统的正常运行。

具体任务如下：

1）从光盘安装 vsftp 软件包，并验证其正确性；

2）配置、编译、安装 tar 格式的 ruby 软件包；

3）应用系统的安装删除软件包检查、安装 Web 应用环境的建立所需要的软件包。

7.3　Linux 系统软件包安装教学目标（4 学时）

1. 教学目标

最终目标：能完成需求软件的安装。

促成目标：

1）知道软件包安装的几种方式；

2）熟悉 RPM 软件包管理软件；

3）熟悉 RPM 软件包安装、查询、删除、升级等方法；

4）知道源代码编译技术。

2. 工作任务

1）查询软件包；

2）安装软件包；

3）删除软件包。

3. 活动设计

4. 相关理论知识

5. 相关实践知识

源文件编译技术。

7.4　项目七　实验指导

7.4.1　RPM 软件包管理工具

（1）软件包查询

rpm-q 软件包名称　　;查询软件包是否安装

$ rpm -q vsftpd　　;查询 vsftpd 软件包是否安装

$ rpm -qa　　;查询所有已安装的软件包

$ rpm -qa grep http　　;查询含有 http 的软件包

$ rpm -qi vsftpd　　;查询软件包的有关说明信息

$ rpm -ql vsftpd　　;查询软件包安装配置目录

（2）删除软件包

rpm-e 软件包名称　　;删除指定软件包

$ rpm -e vsftpd　　;删除 vsftpd 软件包

$ rpm -e vsftpd　　;查询 vsftpd 是否已删除

注意：有的软件包互相有关联，不能简单删除。

（3）软件包安装

首先要安装光盘，挂接光盘文件系统。

1）切换至虚拟机的“视图摘要”，双击 CD-ROM，设置 ISO 文件，并保证设备状态为“已连接”。

2）挂接光盘文件系统

$ mount /dev/cdrom /mnt　　;将光盘挂接到/mnt 目录

3）检查光盘文件系统是否挂接

$ df　　;显示所有挂接的文件系统,若看到挂接点有/mnt 目录,即挂接成功

切换到挂接目录进行安装。

$ cd /mnt/RedHat/RPMS　　;切换至光盘安装目录

$ ls vsftpd *　　;列出 vsftfpd 开头的软件包文件

$ rpm -ivh vsftpd-2.0.1-5.i386.rpm　　;安装软件包,参数 i 表示安装,v 表示显示安装过程,h 显示安装的滚动条

$ rpm -q vsftpd　　;检查 vsftpd 是否已安装

卸载挂接文件系统。

$ cd　　;退出光盘所在目录,返回至/root

$ umount /mnt　　;卸载光盘文件系统

$ df ;检查文件系统是否卸载

（4）软件包升级

rpm -Uvh 软件包全称

7.4.2 源代码软件包安装

（1）获取软件包至安装目录

可以用 ftp 或 samba 等方法，将软件包送到指定目录。

（2）软件包解压缩

根据软件包的压缩格式解压，一般格式为 tar，所以用 tar 命令解压缩。

（3）配置软件包

按安装软件包配置要求进行配置，一般采用 configure 命令。

（4）编译和安装，执行 make 和 make install。

7.5 项目七完成步骤详解

7.5.1 挂接光盘

将光盘放入光盘驱动器，若是虚拟光盘也相同，然后

#mount /dev/cdrom /mnt ;将光盘挂接在/mnt 目录

用 df 命令查看挂接情况（见图 7-1）。

```
[root@fanhs ~]# mount /dev/cdrom /mnt
mount: block device /dev/cdrom is write-protected, mounting read-only
[root@fanhs ~]# df
Filesystem           1K-块      已用      可用  已用% 挂载点
/dev/sda1          4648864   2911992   1500720   66% /
none                130052         0    130052    0% /dev/shm
/dev/hdc              2206      2206         0  100% /mnt
[root@fanhs ~]#
```

图 7-1 查看文件系统挂接情况

7.5.2 安装 vsftpd 软件包

切换到光盘中存放软件包的目录，一般放在光盘的 RedHat/RPMS 目录下。由于光盘挂接在/mnt 目录，所以要切换到/mnt/RedHat/RPMS。

#cd /mnt/RedHat/RPMS

然后查找 vsftpd 软件包，

#ls vsftpd * ;列出所有 vsftpd 开头的文件

进入挂接目录并列出指定文件如图 7-2 所示。

```
[root@fanhs ~]# cd /mnt/RedHat/RPMS
[root@fanhs RPMS]# ls vsftpd*
vsftpd-2.0.1-5.i386.rpm
```

图 7-2　进入挂接目录并列出指定文件

在安装之前，最好首先检查一下是否已安装该软件，

#rpm　-q vsftpd

若已安装，可以先删除，

#rpm　-e　vsftpd

下面开始安装软件包：

rpm　-ihv vsftpd-2. 0. 1-5. i386. rpm　　;安装软件包

安装软件包如图 7-3 所示。

```
[root@fanhs RPMS]# rpm -q vsftpd
vsftpd-2.0.1-5
[root@fanhs RPMS]# rpm -e vsftpd
[root@fanhs RPMS]# ls vsftpd*
vsftpd-2.0.1-5.i386.rpm
[root@fanhs RPMS]# rpm -ihv vsftpd-2.0.1-5.i386.rpm
warning: vsftpd-2.0.1-5.i386.rpm: V3 DSA signature: NOKEY, key ID db42a60e
Preparing...                ########################################### [100%]
   1:vsftpd                 ########################################### [100%]
[root@fanhs RPMS]#
```

图 7-3　安装软件包

最后，仍需要验证一下，是否已安装该软件，

#rpm　-V vsftpd　　;验证是否安装该软件,正确时不显示内容

然后返回到用户根目录，卸载光盘。

#cd　　;返回用户根目录/root

#umount　/mnt　　;卸载光盘

7. 5. 3　安装 tar 格式的源码软件包 ruby

首先通过 ftp 或 smb 方式将 ruby 软件包传送到/root 目录。

将软件包 ruby-1. 9. 1-p0. tar. gz 传送到/root 目录下，在/root 目录下执行解压：

#tar　zxvf　ruby-1. 9. 1-p0. tar. gz

解压至 ruby-1. 9. 1-p0 目录，进入该目录进行配置，

#cd　ruby-1. 9. 1-p0

然后先进行配置：

./configure

生成 makefile 文件，然后进行编译：

#make

最后进行安装：

```
# make  install
```

检验一下：

```
# ruby  --version
```

显示 ruby 版本信息，ruby 安装结束，祝你成功！

7.6 思考题

1）RPM 软件包查询可以查看哪些内容，参数是什么？
2）从光盘安装软件包的基本步骤是怎样的？
3）tar 格式源代码软件包如何安装？

第8讲

基本网络功能应用

 导读

在前面章节中我们学习了网卡的配置、路由设置以及 Ftp 文件传送和 Telnet 远程登录的应用。本章在以上章节的基础上，进一步介绍在 Linux 环境下 NFS 网络协议联网的应用和 Http 网络协议的实现和应用。

8.1 引言

NFS 网络协议使得在 Linux 环境下不同的设备之间的目录能互相挂接，联网后的操作如同在本机目录一样方便，类似于 Windows 系统的网上邻居。Http 网络协议用来实现 Web 站点的设置，本章介绍最基本的 Web 站点的应用，用简单的方法实现网站的应用。

8.1.1 NFS 网络概述

NFS 是网络文件系统（Network File System）的简称，是分布式计算系统的一个组成部分，实现在异种网络上共享和装配远程文件系统。从用户角度看来，在这些远程的文件系统操作和在本地的文件系统上操作并没有什么不同。NFS 由 Sun 微系统公司（Sun Microsystem，Inc）开发，制订了 NFS 标准，并被 IETF 接受，纳入 RFC，作为文件服务的一种标准（RFC1904，RFC1813）。NFS 基于客户/服务器结构，通过 RPC（远过程调用）实现，所有的 NFS 操作都由 RPC 过程来进行。NFS 服务器导出本地的目录给远程的 NFS 客户，NFS 客户把对文件操作系统调用重定向到远程的系统。在 Linux 里，通常用 knfsd 来实现 NFS 服务，这是个运行在核心空间的后台守护程序，相对于用户空间的 NFS 程序，有较高的响应性能。

NFS 是目前最典型的采用了 RPC 的网络服务系统。它允许用户完全像访问自己的本地文件一样，访问远程主机上的文件。这是通过综合客户机端上的内核功能（使用远程文件系统）和服务器端上的 NFS 服务器（提供文件数据）来实现的。对客户机来说，这种文件访问完全是透明的，而且是通过不同的服务器和主机架构来实现的。

8.1.2 NFS 的优点

1）占据大量磁盘空间的数据也可保存在一台单一的主机上。比如，所有与 LaTeX 和 METAFONT 相关的文件和程序可集中在同一个地方保存和维护。

2）管理性数据也可保存在一台单一的主机上。要在 20 台机器上安装同一个文件时，不再需要 rpc。NFS 中的大部分都是 RickSladkey（邮件地址 jrs@ world. std. com）编写的，他还编写了 NFS 内核代码的 Linux 实施方案和大部分 NFS 服务器。NFS 服务器源于 unfsd 用户空间 NFS 服务器和 hnfsHarrisNFS 服务器，前者最初是由 MarkShand 编写的，后者则是由 DonaldBecker 编写的。

3）供所有用户访问的数据可保存在一台中央主机上，并在启动时随客户装载这个目录。举例来说，可将所有用户的账号信息保存在一台主机上，令网络上的所有主机都从它那里装入 /home。依靠已安装了的 NIS，用户就可以登录到任何系统，采用的仍是同一个文件集。

8.1.3 NFS 工作原理

先来看看 NFS 的工作原理：客户机可以像安装物理设备一样，要求在本地目录上安装远程主机内的目录。但是用于指定远程目录的句法有所不同，比如，为了将 vlager 上的/home 目录安装到 vale 主机上的/users，管理员将在 vale 上执行下面的命令（可省略-tnfs 参数，因为 mount 能够从冒号得知它指定了一个 NFS 卷）：

#mountvlager：/home/users

然后，mount 将试着通过 RPC 链接到 vlager 主机上的 mountd 安装程序。服务器便查看 vale 是否被许可安装这个目录，如果可以，就为 vale 返回一个文件句柄。后续发出的请求/users 目录下的所有文件，都将采用这个句柄。

当有人通过 NFS 访问文件时，内核会任命一个指向服务器主机上的 nfsd（NFSdaemon）的 RPC 调用。这个调用将采用前面的文件句柄、准备访问的文件名和用户的用户 ID 以及组 ID 作为自己的参数。采用这些参数的目地是判断用户是否被许可访问指定文件。为了防止未授权用户读取和修改文件，两台主机上的用户 ID 和组 ID 必须是一样的。

在多数实施中，客户机和服务器的 NFS 机器都是作为内核级程序来实施的，这种内核级程序在系统启动时，便从用户空间开始运行。它们就是服务器主机上的 NFSdaemon（nfsd）和客户机上运行的 BLOCKI/ODaemon（biod）。为了改进网络流通量，boid 执行了先读后写方式的异步 I/O 数据块，并且，若干个 nfsd 程序可同步运行。

NFS 实施和客户机代码紧密集成在内核的虚拟文件系统（VFS）层相比稍有不同，它不需要通过 boid 进行额外的控制。另一方面，服务器代码完全在用户空间内运行，所以同时运行多个服务器备份将由于涉及同步问题，几乎是不可能的（请注意，NFS 的内核版正在开发过程中）。目前，NFS 还缺乏先读后写机制，但有幸的是，RickSladkey 正在筹划此事（后写这个问题是指内核缓冲区按 device/inode 对制作索引，因此，不能用于 NFS 安装文件系统）。

关于 NFS 代码，最大的问题是内核版本 1 分配的内存块不得大于 4K，不这样的话，将导致连网代码不能对有些大型数据报进行处理，这些数据报在减掉报头等之后，字节数仍然超过 3500。也就是说，对运行于这些系统（它们采用的是默认设置大型 UDP 数据报，例如 SunOS 系统采用的是 8K）上的 NFSdaemon 来说，要对它们进行投递，需要将它们人工地拆成小的数据包。某些情况下，这样做可能导致性能急剧降低（AlanCox 是这样解释的：NFS 规范要求在服务器返回应答之前，将每次写操作填入（flush）磁盘。由于 BSD 内核只能接受 4K 的写操作，所以将 4 个 1K 大小包写入基于 BSD 的 NFS 服务器，将产生 4 个 4K 大小的写操作）。这个问题在后来的内核 1.1 版本中得以解决，而且还对客户机代码进行了修

改，使之可以利用这一特性。

8.1.4 Web 服务概述

Internet 上最热门的服务之一就是环球信息网 WWW（World Wide Web）服务，Web 服务已经成为许多人上网查找、浏览信息的主要手段。WWW 是一种交互式图形界面的 Internet 服务，具有强大的信息链接功能。它使得成千上万的用户通过简单的图形界面就可以访问各大学、组织、公司等组织机构和个人的最新信息和各种服务。

商业界很快看到了其价值，许多公司建立了主页，利用 Web 在网上发布消息，并将它作为各种服务的界面，如客户服务、特定产品和服务的详细说明、宣传广告以及日益增长的产品销售和服务。商业用途促进环球信息网络迅速发展。

Web 服务具有如下特点：

1）Web 是图形化的和易于导航的；

2）Web 是与平台无关的；

3）Web 是分布式的；

4）Web 是动态的；

5）Web 是交互的。

Web 系统是客户/服务器模式的。所以应该有服务器端程序和客户端程序两部分。常用的服务器程序软件是 Apache，常用的客户端程序软件是浏览器（如 IE、Netscape、Mozila 等）。我们可以在浏览器的地址栏内输入统一资源定位地址（URL）来访问 Web 页面。Web 最基本的概念是超文本（Hypertext）。它使得文本不再是传统的书页式文本，而是可以在阅读过程中从一个页面跳转到另一个页面位置。用来书写 Web 页面的语言称为超文本标记语言，即 HTML。WWW 服务遵从 HTTP 协议，默认的 TCP/IP 端口是 80。

8.1.5 Apache 服务器

根据著名的 WWW 服务器调查公司所做的调查，世界上百分之五十以上的 WWW 服务器都在使用 Apache，是世界排名第一的 Web 服务器。选择 Web 服务器时，其功能和运行性能是最重要的因素。Apache 的众多特性保证了它可以高效而且稳定的运行。其性能主要表现在如下方面：

1）实现动态共享对象（DSO），允许在运行时动态装载功能模块；

2）采用预生模块的技术提高响应速度；

3）可以运行在几乎所有的计算机平台；

4）支持最新的 HTTP1.1 协议；

5）简单而强有力的基于文件的配置；

6）支持虚拟主机；

7）支持 HTTP 认证；

8）集成了代理服务器；

9）具有可定制的服务器日志；

10）支持安全 Socket 层（SSL）；

11）用户会话的跟踪能力；

12）支持通用网关接口（CGI）；

13）集成 Perl 脚本编程语言；
14）支持服务器端包含命令（SSI）；
15）支持 FastCGI；
16）支持 PHP；
17）支持 Java Servlets；
18）支持第三方软件开发商提供的大量功能模块。

运行 Apache 不需要太多的计算资源。它在有 6 ~ 10MB 硬盘空间和 8MB RAM 的 Linux 系统上运行得很好。然而，只运行 Apache 可能不是你想做的事情。更可能的是，你想运行 Apache 来提供 WWW 服务、启动 CGI 进程以及充分利用所有 WWW 能够提供的令人惊奇的功能。在这种情况下，你需要提供反映负载要求的额外的磁盘空间和内存空间。也就是说，如果仅仅是启动 WWW 服务并不需要太多的系统资源，但是想要能为大量的客户提供服务就需要更多的系统资源。

8.2 企业项目场景 8：基本网络功能应用

在公司内部实现两种联网方式：
1）NFS 网络协议联网，实现文件共享、传输操作；
2）HTTP 网络协议应用，实现 Web 网站的浏览访问。
NFS 网络协议联网具体任务：

表 8-1 NFS 共享目录

目录说明	NFS 共享目录	权限	文档
凭证汇总	/home/cwb/Pz	本部门可读可写，其他部门无权使用	Pz1. txt Pz2. txt Pz3. txt
报表汇总	/home/cwb/Bb	本部门可读可写，其他部门可读	Bb1. txt Bb2. txt
通知汇总	/home/cwb/Tz	本部门可读可写，其他部门可读	Tz1. txt Tz2. txt
其他	/home/cwb/Qt	全公司可读可写	Qt. txt

Web 网站开通要求：
1）网站域名为部门 . net. cn，如 cwb. net. cn；
2）网站主页为 index. html。

8.3 基本网络功能应用教学目标（8 学时）

1. 教学目标

最终目标：能用 NFS 协议实现 Linux 系统网络共享。开通 Web 网站，并能浏览该网站。
促成目标：

1）NFS 网络的应用技术；
2）Web 网站启用技术。

2. 工作任务

1）设置 NFS 网络；
2）设置基本 Web 网站；
3）应用 NFS 网络协议联网；
4）在 Windows 操作系统浏览验证 Web 网站。

8.4　项目八　实验指导

8.4.1　NFS 网络

（1）建立共享目录

```
#mkdir  /var/testnfs                ;将/var/testnfs 作为共享目录
#chmod  777  /var/testnfs           ;将该目录存储权限放开至最大
```

（2）设置 nfs 配置文件

```
#vi  /etc/exports                   ;编辑 nfs 配置文件,加入
/var/testnfs      *(rw,sync)        ;加入共享目录,并允许所有外部设备可以访问,具
                                     有可读可写,同步写入的权限。rw(可读可写),ro
                                     (只读),sync(同步),可以将 * 改为具体的机器名
                                     或 IP 地址。
```

（3）启动 nfs 服务，检查 nfs 输出情况

```
#service nfs start                  ;启动 nfs 网络服务
#showmount  -e  NFS 服务器或 IP 地址
```

（4）挂接 nfs 网络文件

```
mount  -t  nfs-o  rw  服务器地址:挂接目录  本地挂接目录
#mount  -t nfs-o rw 192.168.1.100:/var/testnfs  /mnt   ;/mnt 就是联网目录,在/mnt
                                                         读写就是在对方目录读写。
                                                         挂接后,本机所有用户都可
                                                         以访问。
```

（5）卸载网络

```
#umount  /mnt                       ;卸载网络
```

8.4.2　Web 网络

（1）查看 httpd 软件包是否安装

```
#rpm  -q httpd                      ; 查看软件包
```

（2）启动 httpd 服务

```
#service httpd start                ; 启动 Web 服务
#service httpd status               ; 检查 httpd 进程是否运行
```

（3）编辑网页

#cd /var/www/html ；切换目录到默认网页目录

#vi index. html ；编辑主页 index. html

（4）浏览网页

打开浏览器，输入 http：//主机 IP 地址。可以在 Windows 环境下浏览，只要网络是联通的即可。

8.5 项目八完成步骤详解

8.5.1 NFS 网络实现

1. 在所有设备上设置简单域名

```
#vi  /etc/hosts
192.168.1.200    cwb     ;财务部主机名
192.168.1.210    cw1     ;财务部 1 主机名
192.168.1.220    cw2     ;财务部 2 主机名
192.168.1.230    cw3     ;财务部 3 主机名
192.168.1.240    yw1     ;业务部 1 主机名
```

○○○○○○

2. 编辑 NFS 共享设置文件/etc/exports

cw1、cw2、cw3 为财务部所属机器名，编辑/etc/exports 文件，设置财务部目录凭证、报表、通知、其他（/home/cwb/Pz，/home/cwb/Bb，/home/cwb/Tz，/home/cwb/Qt）4 个目录的访问权限，财务部所属机器对上述目录都是可读可写，其他部门为只读，仅有其他目录为可读可写。设置 NFS 配置文件如图 8-1 所示。

vi /etc/exports ；在财务部主机上设置 NFS 配置文件

```
/home/cwb/Pz     cw1(rw,sync) cw2(rw,sync) cw3(rw,sync)
/home/cwb/Bb     cw1(rw,sync) cw2(rw,sync) cw3(rw.sync)
/home/cwb/Tz     cw1(rw,sync) cw2(rw,sync) cw3(rw,sync)

/home/cwb/Pz     *(ro,sync)
/home/cwb/Bb     *(ro,sync)
/home/cwb/Tz     *(ro,sync)
/home/cwb/Qt     *(rw,sync)
```

图 8-1 设置 NFS 配置文件

3. 在财务部主机上建立财务部账户及目录

```
#groupadd  cwb            ;建立财务部组 cwb
#useradd  -g  cwb  cwb    ;在财务部组中建立账户 cwb
#passwd  cwb              ;设置口令
#su  - cwb                ;切换到 cwb 用户账户
$ mkdir Pz  Bb  Tz  Qt    ;建立财务部目录凭证、报表、通知和其他
```

$ chmod 777 *　　　;开放所有目录,只在/etc/exports 文件中设置权限

$ chmod 777 /home/cwb　　;开放财务部目录(否则下层目录不能进入)

在财务部主机上启动 NFS 服务:

#service nfs start　　　;启动 NFS 服务

#iptables -F　　　;关闭防火墙

4. 财务部本部 NFS 共享

将 cw1 设备连接至财务部主机:

#showmount -e cwb　　　;检查 cwb 主机 NFS 共享情况(见图 8-2)

```
[root@fanhs ~]# showmount -e cwb
Export list for cwb:
/home/cwb/Qt *
/home/cwb/Tz (everyone)
/home/cwb/Bb (everyone)
/home/cwb/Pz (everyone)
[root@fanhs ~]#
```

图 8-2　检查 cwb 主机 NFS 共享情况

mount -t nfs-o -rw cwb:/home/cwb/Qt /mnt;挂接财务部 Qt 目录至/mnt (见图 8-3)

```
[root@fanhs ~]# mount -t nfs -o rw cwb:/home/cwb/Qt  /mnt
[root@fanhs ~]# df
Filesystem          1K-块      已用       可用  已用% 挂载点
/dev/sda1          4648864   2912012   1500700   66% /
none                130052         0    130052    0% /dev/shm
cwb:/home/cwb/Qt   4648864   3112192   1300512   71% /mnt
[root@fanhs ~]#
```

图 8-3　挂接财务部 Qt 目录至/mnt

将其他目录也一起挂接，但需要在/mnt 下建立相应目录，如图 8-4 所示。

```
[root@fanhs ~]# mkdir /mnt/Pz
[root@fanhs ~]# mkdir /mnt/Bb
[root@fanhs ~]# mkdir /mnt/Tz
[root@fanhs ~]# mount -t nfs -o rw cwb:/home/cwb/Pz /mnt/Pz
[root@fanhs ~]# mount -t nfs -o rw cwb:/home/cwb/Bb /mnt/Bb
[root@fanhs ~]# mount -t nfs -o rw cwb:/home/cwb/Tz /mnt/Tz
[root@fanhs ~]# df
Filesystem          1K-块      已用       可用  已用% 挂载点
/dev/sda1          4648864   2912012   1500700   66% /
none                130052         0    130052    0% /dev/shm
cwb:/home/cwb/Qt   4648864   3112224   1300512   71% /mnt
cwb:/home/cwb/Pz   4648864   3112224   1300512   71% /mnt/Pz
cwb:/home/cwb/Bb   4648864   3112224   1300512   71% /mnt/Bb
cwb:/home/cwb/Tz_  4648864   3112224   1300512   71% /mnt/Tz
```

图 8-4　其他目录也一起挂接

Qt 目录也需要同样处理，这样我们就可以在/mnt 目录下处理财务部的所有文件了，就像在本地目录一样。

在 cw1 共享目录使用测试：

#cd /mnt/Pz ；切换到凭证目录

#touch pz1. txt pz2. txt pz3. txt ；新建 3 个文件（见图 8-5）

```
[root@fanhs ~]# su - cwb
[cwb@fanhs ~]$ cd /mnt/Pz
[cwb@fanhs Pz]$ touch pz1.txt pz2.txt pz3.txt
[cwb@fanhs Pz]$ ls
pz1.txt  pz2.txt  pz3.txt
[cwb@fanhs Pz]$ rm pz3.txt
[cwb@fanhs Pz]$ ls
pz1.txt  pz2.txt
[cwb@fanhs Pz]$
```

图 8-5 新建 3 个文件

超级用户挂接目录后，不同用户可以使用目录。在 cw1 设备中，用户可以在凭证目录中添加、删除文件。

5. 其他部门 NFS 共享

将业务部 yw1 设备连接至财务部主机：

#showmount -e cwb ；检查 cwb 主机 NFS 共享情况（见图 8-6）

```
[root@fanhs ~]# showmount -e cwb
Export list for cwb:
/home/cwb/Qt *
/home/cwb/Tz (everyone)
/home/cwb/Bb (everyone)
/home/cwb/Pz (everyone)
[root@fanhs ~]#
```

图 8-6 检查 cwb 主机 NFS 共享情况

建立挂接目录：

#mkdir /mnt/Qt /mnt/Pz /mnt/Bb /mnt/Tz

挂接 NFS 服务器财务部目录：

#mount-t nfs-o rw cwb:/home/cwb/Qt /mnt/Qt ;挂接其他目录,可读可写

#mount-t nfs-o ro cwb:/home/cwb/Pz /mnt/Pz ;挂接凭证目录,只读

#mount-t nfs-o ro cwb:/home/cwb/Bb /mnt/Bb ;挂接报表目录,只读

#mount-t nfs-o ro cwb:/home/cwb/Tz /mnt/Tz ;挂接通知目录,只读

注意与财务部设备挂接的区别。

分别将财务部的文件夹挂接到指定目录如图 8-7 所示。

```
[root@fanhs ~]# mkdir  /mnt/Qt  /mnt/Pz  /mnt/Bb  /mnt/Tz
[root@fanhs ~]# mount -t nfs -o rw  cwb:/home/cwb/Qt  /mnt/Qt
[root@fanhs ~]# mount -t nfs -o ro  cwb:/home/cwb/Pz  /mnt/Pz
[root@fanhs ~]# mount -t nfs -o ro  cwb:/home/cwb/Bb  /mnt/Bb
[root@fanhs ~]# mount -t nfs -o ro  cwb:/home/cwb/Tz  /mnt/Tz
[root@fanhs ~]# df
Filesystem           1K-块      已用      可用  已用% 挂载点
/dev/sda1           4648864   2912004   1500708  66% /
none                 130052         0    130052   0% /dev/shm
cwb:/home/cwb/Qt    4648864   3112224   1300512  71% /mnt/Qt
cwb:/home/cwb/Pz    4648864   3112224   1300512  71% /mnt/Pz
cwb:/home/cwb/Bb    4648864   3112224   1300512  71% /mnt/Bb
cwb:/home/cwb/Tz    4648864   3112224   1300512  71% /mnt/Tz
```

图 8-7　分别将财务部的文件夹挂接到指定目录

在 yw1 共享目录使用测试（见图 8-8）。

```
[root@fanhs ~]# groupadd ywb
[root@fanhs ~]# useradd -g ywb ywb1
[root@fanhs ~]# su - ywb1
[ywb1@fanhs ~]$ cd /mnt/Pz
[ywb1@fanhs Pz]$ ls
pz1.txt  pz2.txt
[ywb1@fanhs Pz]$ touch pz3.txt
touch: cannot touch ‘pz3.txt’: 只读文件系统
[ywb1@fanhs Pz]$ rm pz1.txt
rm: 无法删除 ‘pz1.txt’: 只读文件系统
```

图 8-8　共享目录使用测试

可以看到文件，但不能建立和删除文件，符合设计要求。

注意：在这里即使将挂接属性设置成 rw 也是不行的，因为在服务器端已经设置为 ro 了。

8.5.2　Web 网络实现

1. 确认 httpd 软件包已经安装

确认 httpd 软件包已经安装如图 8-9 所示。

```
[root@cwb ~]# rpm -q httpd
httpd-2.0.52-9.ent
```

图 8-9　确认 httpd 软件包已经安装

2. 启动 httpd 服务，并检查运行状态

检查运行状态如图 8-10 所示。

```
[root@cwb ~]# service httpd start
启动 httpd:                                                [  确定  ]
[root@cwb ~]# service httpd status
httpd (pid 3565 3564 3563 3562 3561 3560 3559 3558 3555) 正在运行...
```

图 8-10　检查运行状态

3. 在默认路径编辑网页

编辑网页如图 8-11 所示。

```
[root@cwb html]# cd /var/www/html
[root@cwb html]# vi index.html
[root@cwb html]# cat index.html
<h1> This is CWB Website !!! </h1>
```

图 8-11　编辑网页

4. 用浏览器浏览网页

浏览网页如图 8-12 所示。



图 8-12　浏览网页

也可以用简单域名浏览，如图 8-13 所示。



图 8-13　用简单域名浏览

8.6　思考题

1）如何设置 NFS 网络共享？
2）如何设置和启用 Web 网站？
3）如何上载和下载文件？
4）如何实现远程登录（参见项目二）？
5）若服务已正确开启，但却不能联网到服务器，可能是什么问题？
（提示：从安全角度考虑）

第9讲 Shell 程序设计基础

9.1 引言

登录 Linux 系统之后，就会出现一个系统提示符，如#或 $ 。可以在提示符后输入指令，并获得预期的结果。是什么帮我们做了这一切呢？就是 Shell（壳）！其实，当登录系统时，就进入了一个 Shell 中，它担任了翻译的角色，将输入的每一个指令进行解读和处理，所以也将 Shell 称为直译器。

Shell 本身也是 Unix 中的一个程序，所以，你可以依照你的喜好选择不同的 Shell 来担任你的命令直译器。目前，常见的 Unix Shell 有两大主流：一是 Bourne Shell(/bin/sh)，衍生的 Shell 有 GNU Bourne Again Shell(/bin/bash)、ash、ksh、zsh 等；另一支主流是模仿 C 语言格式的 C Shell(/bin/csh)，而衍生的 tsch(Tenex C Shell) 更是众多 Unix 行家青睐的 Shell。有这么多的 Shell，你随时都可以使用你想使用的 Shell。当登录 Linux 系统时，系统内定使用 bash。

Shell 除了帮助用户解析并处理命令外，还有七大重要特性，它是辅助你玩转 Linux 系统的利器。不过众多 Shell 类别，各种 Shell 间的特性虽然相近，设置方法却不尽相同。如 bash 使用 export 设置环境变量，tcsh 却使用 setenv 设计环境变量。这里以 bash 为主轴进行介绍，读者可以触类旁通。

1. Shell 交谈模式（interactive）

当登录系统（进入 Shell）后，Shell 就会显示系统提示符一直等待用户输入命令，当 Shell 接受到输入的命令后，就开始执行这项命令并把结果返回到屏幕上。在结束这个命令后，Shell 又会呈现系统提示符等待用户的下一个命令，直到用户执行 exit 或按 Ctrl + d 结束 Shell 退出系统后方休。上述操作的直觉就是 Shell 的交谈模式。

另外，Shell 还具有 History（历史）功能，按方向键的“上键”可重复上一个操作。也可执行 history 指令查看曾经执行过的指令，Linux 的 bash 内定记录 1000 个以前曾经执行过的指令，并将每个执行过的指令编号，可以用“！编号”重新执行该指令。

除了 history 功能外，交谈模式还有一个 File Completion 功能，就是对很长的文件名，只要输入前面几个字符，然后按 Tab 键，系统就会自动找到符合要求的文件名并补齐。若符合要求的文件名不止一个，Shell 就会列出所有符合要求的文件名，继续输入字符，直到可以唯一地区分出文件名，系统将自动补齐。

2. 后台执行（run commands in the background）

使用 Shell 交谈模式一次仅可以完成一件工作，当同时想完成几件事情以发挥 Linux 的多工作能力时，又不想开设多个终端（terminal），Shell 提供了绝佳的解决方案——后台执行。后台执行的符号是“&”，在命令行输入命令时，在最后加上 &，系统就会将程序放在后台执行，并显示提示符等待输入下一个命令，而不必等待该命令执行完。

若程序已经在执行了，又想让该程序转为后台执行，该如何操作呢？首先按 Ctrl + z 暂停程序，然后按 bg 将暂停的程序转入后台执行即可。

3. 输入输出重定向（input/output redirection）

可以使用输入导向“ < ”或输出导向“ > ”将标准输入装置（stdin）键盘或标准输出装置（stdout）显示器的输出重新导向文件、打印机或其他装置如/dev/null。

4. 管道（pipes）

管道的功能是组合一系列的指令按顺序进行处理，并将上一程序的输出作为下一个程序的输入，依次执行。各种指令通过管道的组合可以完成复杂的功能。

5. 通配符（wild-card characters）

通配符可以让你在处理文件时顺利地找到匹配的文件或其他特殊组合的文件。例如，* 代表任何字符（可以是零个、一个或多个），? 代表任何一个字符。

6. 环境变量（environment variables）

Shell 是个庞大的程序，为了使工作环境符合我们的习惯或需要，Shell 提供了环境变量功能，让你自己设计 Shell 使用环境。要查看环境变量的方法是利用 $ 符号加上变量名称，在命令行输入 $ HOME、$ PATH，就会看到这些环境变量的内容。也可以直接执行 export 指令，显示目前系统所有的 Shell 环境变量。

7. Shell 脚本

Shell 最重要也是最复杂的功能就是 Shell 脚本（Shell script），Shell 本身除了负责解析命令外，还有一个重要的身份就是编程语言（Programming language）。简易的 Shell 脚本就是将一堆指令集合在一起，顺序执行以完成某些功能，有点类似于 DOS 的批处理文件（batch file）。

9.2 Shell 程序设计基础教学目标（8 学时）

1. 教学目标

最终目标：掌握 Shell 程序设计的基本方法。

促成目标：

1）掌握 Shell 变量、环境变量的设置；

2）Shell 执行的几种方法；

3）熟悉 Shell 程序设计的常用语句。

2. 工作任务

1）设置环境变量；

2）编写最简单的 Shell 程序并运行；

3）用分支语句、循环语句编写 Shell 程序。

3. 活动设计

编写一个加法程序，分别用参数形式输入和文件输入形式实现。

4. 相关理论知识

5. 相关实践知识

9.3 Shell 程序设计基础　实验指导

9.3.1 Linux 变量

（1）简单变量

```
$ color = red                                 ;设置变量 color 的值为 red
$ echo $ color                                ;显示变量 color 的值
  red
$ echo I like color $ color                   ;在语句中显示变量值
  I like color red
$ echo I like $ {color} Hat                   ;用大括号分隔
  I like redHat
```

（2）列表变量

```
$ declare -a team                             ;定义数组
$ team[0] = "Beijing"                         ;赋值
$ team[1] = "Shanghai"
$ team[2] = "NewYork"
$ echo $ {team[0]} $ {team[1]} $ {team[2]};显示
Beijing Shanghai NewYork
```

（3）导出变量（环境变量）

在所有的 Shell 中，变量都是本地的局部变量，除非另外指定，不会传递给子 Shell。export 命令能把局部变量变成环境变量，相当于全局变量。

```
$ export                                      ;列出当前导出变量
$ DATABASE = /dbbase/db                       ;新建一个局部变量
$ export DATABASE                             ;导出 DATABASE
$ env                                         ;列出环境变量
DATABASE = /dbbase/db                         ;发现该变量已加入
............
```

切换至子程序，该变量依然存在（如 su user1，查看 env）。

（4）只读变量

```
$ password = abcd                             ;定义一个局部变量
$ readonly password                           ;设置只读
$ readonly                                    ;列出只读变量
```

```
declare  -r  password ="abcd"
$ password =12345                          ;重新赋值
bash:password:  readonly variable          ;指示该变量已只读
```

9.3.2 重定向和管道

（1）重定向

重定向就是将输出方向改变。例如，原来输出到屏幕，现在改为输出至文件中。

```
$ ls  -l                                   ;详细列表输出至屏幕
$ ls  -l > ls.txt                          ;改为输出至 ls.txt 文件中,屏幕不再显
                                            示,ls.txt 文件若存在将被覆盖
```

（2）追加

```
$ ls  -l >> ls.txt            ; 追加内容至文件后部,若 ls.txt 文件不存在将新建
```

（3）管道

将上一个程序执行的结果传给下一个程序，就像流水线一样，将结果往下传递。

例如，找出进程中有关 user1 的进程：

```
ps aux | grep  user1          ; 首先执行 ps，然后在结果中查找 user1 的相关项
```

9.3.3 命令替换

看一个实例：

```
$ echo there are $ (who | wc-l) users on the system
there are 3 users on the system
```

$ ( ) 也可以用抑音括起符号''代替：

```
$ echo there are  'who | wc -l'  users on the system
```

效果相同，且更简捷。

9.3.4 算术运算

((operation))

其中 operation 可以是：

+ −	加、减
++ −−	增量、减量
* / %	乘、除、求余
**	求幂

例如：

```
$ a =3 ; ((a ++)); echo $ a
4
```

9.3.5 Shell 程序执行和程序参数

（1）Shell 程序编辑和执行

Shell 程序是一个文本文件，所有的系统命令都可以作为语句。因此，用任何文本编辑

工具就可以编程了。在 Linux 环境下，最常用的编辑工具就是 Vi 编辑器。Shell 程序文件的后缀也没有要求，为了区分文件类别，常常将后缀写作 . sh。

（2）Shell 程序的执行

Shell 程序的执行有这样几种方法：

设要执行一个 Shell 程序 test. sh

```
$ sh    test. sh              ;用 sh 或 bash 命令执行
$ .     test. sh              ;用 . (句号)执行
$ source    test. sh          ;用 source 命令执行
```

若需要直接执行 Shell 程序文件，则需要符合两个条件：

1）Shell 程序文件本身可执行

```
$ chmod    a + x    test. sh ;给 test. sh 文件加上可执行权限
```

2）应将 Shell 程序放在系统设定的程序目录，如/bin　/usr/bin　/home/user/bin 等满足以上条件的 Shell 程序，即可直接打程序名执行。

（3）Shell 程序执行参数（位置参数）

$#：传给 Shell 程序的参数个数；

$ *：传给 Shell 程序全部参数组成的字符串；

$ 0：Shell 程序名（第 0 个参数）；

$ 1：第 1 个参数；

$ 2：第 2 个参数；

…………………………

$?：程序执行的返回值。

例如：

```
$ cat    a. sh
echo  $ #
echo  $ 1  $ 2  $ 3  $ 4  $ 5
echo  $ 0
$ sh    a. sh    a    b    c    d    e
5
a    b    c    d    e
a. sh
```

参数的位置可以用 shift 命令后移和前移。

9. 3. 6　条件表达式

（1）算术测试

运算符：

```
< = 、> = 、< 、>              ;小于等于、大于等于、小于、大于
= = 、!  =                    ;相等、不相等
!                             ;逻辑非
&&                            ;逻辑与
```

||　　　　　　　　　　　　　　;逻辑或

算术测试用(()),例如:

If(($ result == 0))

While(($ count <= $ testval));do

(())内变量名前可以不用 $ 。

(2) 字符串比较

字符串条件运算符:

-n string　　　　　　　　;字符串长度不为0

-z string　　　　　　　　;字符串长度为0

sring1 == string2　　　　;字符串相等为真

string1 ! = string2　　　;字符串不相等为真

例如:

```
$ a=abc
$ if [ -n $ a ] ; then echo a is not empty ; fi
a is not empty
```

注意:字符串比较用方括号,括号、变量、运算符之间必须有空格,否则出错!

(3) 面向文件表达式

-e file　　　　　　;若文件存在则为真

-f file　　　　　　;若文件存在且是普通文件为真

-r file　　　　　　;若文件存在且可读为真

-w file　　　　　　;若文件存在且可写为真

-x file　　　　　　;若文件存在且可写为真

-d file　　　　　　;若文件存在且是目录则为真

例如:

```
$ mkdir a
$ if [ -d a ] ; then echo a is a directory ; fi
a is a directory
```

测试也是用 [ ],且与变量符号之间必须有空格分隔。

9.3.7 控制结构

(1) if..then..elif..then..else fi 语句

```
$ cat if.sh
echo "Please input n1,n2:"
read n1 n2
if((n1 ==n2))
then
   echo "n1 =n2 "
elif((n1 >n2))
then
```

```
   echo "n1 > n2 "
elif((n1 < n2))
then
   echo "n1 < n2 "
else
   echo "n1 < > n2 "
fi
```

(2) case. . in. . esac 语句

```
 $ vi  testcase. sh
team[1] = "shanghai"
team[2] = "ningbo"
team[3] = "hangzhou"
team[4] = "wengzhou"
select = 1
while((select > 0))
do
   echo 1   shanghai
   echo 2   ningbo
   echo 3   hangzhou
   echo 4   wengzhou
   echo 0   quit
   echo   your choise?
   case $ {team[ $ select]}   in
      "shanghai")echo you select shanghai team ;;
      "ningbo")   echo you select niingbo team ;;
      "hangzhou")echo you select hangzhou team ;;
      "wengzhou")echo you select wengzhou team ;;
   0)   exit ;;
      * )           echo you select unknown team ;;
   esac
done
```

(3) for. . do. . done 语句

例 1,将所有参数相加:

```
 $ cat   for1. sh
sum = 0
for i in $ *
do
   ((sum + = i))
   echo $ i $ sum
```

```
done
$ sh for1. sh 1 2 3 4 5
  1   1
  2   3
  3   6
  4   10
  5   15
```

例 2,将文件中的数相加：

```
$ cat   a. txt
1   2   3   4   5
$ cat for2. sh
for i in 'cat a. txt '
do
  ((sum + =i))
  echo $ i   $ sum
done
```

结果同例 1。

(4) while. . do. . done

例如,将 1 ~100 中偶数相加：

```
$ cat while. sh
sum =0
n =0
while((n < =100))
do
  ((n ++ ))
  if((n%2 ! =0))
    then
    continue
    fi
    ((sum + =n))
    echo $ n   $ sum
done
echo The total number is $ sum
```

(5) untile. . do. . done

与 While. . do. . done 条件判断相反，until. . do. . done 是条件为假时执行，条件为真时结束。例如，每隔 1s 打一个点，共 10 个点：

```
$ cat untile. sh
n =1
until((n >10))
```

```
do
   echo -n ". "
   sleep 1
   ((n ++))
done
echo
```

9.3.8 函数

可以在 Shell 程序内设置函数。例如，设置一个加法函数并调用：

```
$ cat fun1. sh
f(){
   ((value = $ 1 + $ 2))
   return   $ value
   }
   f  10  20
   echo return value from function was  $ ?
    $ sh fun. sh
   return value from function was 30
```

(1) 递归

用函数递归计算阶乘：

```
$ cat fact. sh
fact(){
   if(( $ 1 < =1))
   then
      echo 1
   else
      typeset tmp
      typeset result
      ((tmp = $ 1-1))
      ((result ='fact  $ tmp' *  $ 1))
      echo  $ result
   fi
}
echo "fact(8) ='fact 8 '"
$ sh fact. sh
fact(8) =40320
```

注意：函数中的返回要用 echo $ value！

(2) 共享函数

可以先把函数读到内存空间，然后就可以不断调用，无需将函数写入本程序内。

设置一个加法函数：

```
$ cat add. sh
add( )  {
    ((value = $ 1 + $ 2))
    echo $ value
}
```

再做一个测试程序：

```
$ cat test. sh
source  add. sh            ;将函数读入内存
b = $ (add 100 200)
echo $ b
```

执行程序：

```
$ sh test. sh
300
```

注意：读函数只能用 source 或 . （句号），不能用 sh。

9. 3. 9 菜单命令 select

用 select 命令与 case 语句配合，可以快速组成菜单：

```
$ cat select. sh
echo menu test program
select reply in 1 2 3
do
  case $ reply in
      1)date ;;
      2)pwd ;;
      3)break ;;
      4)echo illegal choise ;;
  esac
done
```

9. 3. 10 作业控制

对执行的程序进行控制，可以让程序在前台、后台执行和删除程序的执行。

```
jobs                ;显示当前 Shell 的所有作业
bg      %n          ;让 n 号作业在后台执行
fg      %n          ;让 n 号作业在前台执行
kill    %n          ;删除 n 号作业
```

操作实例：

```
$ vi test. sh       ;编辑 test. sh 文件,然后按 ctrl + z,停止操作
$ jobs              ;列出当前作业
```

```
[1]+   Stopped      vi   test.sh
$ fg   %1                ;恢复该作业,然后再次按 Ctrl+z
[1]+   Stopped      vi test.sh
$ kill   %1              ;删除该作业
[1]+   已终止       vi   test.sh
```

第10讲 C 语言编程工具

10.1 引言

C 语言作为一种当前使用非常广泛的高级程序设计语言，具有简单易用、跨平台、可移植性好的特点。Linux 内核就是使用 C 语言开发的。C 语言是 Linux 上的主要开发语言，它在 Linux 编程开发中扮演着重要的角色，它们形成了相得益彰的完美组合，为用户提供了一个强大的编程环境。在国内很多 Linux 爱好者仅停留在系统管理的层次上，而更多程序员要做 Linux 下的程序开发却无从下手，重要的是况且很难找到合适的学习参考资料。这本书正是从这样的结合点出发，介绍在 Linux 系统中使用 C 语言编程的有关知识。读者通过本书的学习能够快速地学会 Linux 下 C 语言的编程，掌握其中的编程方法和技巧，并能从一开始就养成良好的编程习惯，从而实现 Linux 环境下的编程知识入门和提高。

C 语言是一种在 Unix 操作系统中早期就被广泛使用的通用编程语言。它最早是由贝尔实验室的 Dennis Ritchie 为了 Unix 的辅助开发而写的，开始时 Unix 是用汇编语言和一种叫 B 的语言编写的。从那时候起，C 语言就成为世界上使用最广泛的计算机语言。

C 语言能在编程领域得到如此广泛的支持原因有以下一些：

1）它是一种非常通用的语言。几乎任何一种计算机上都有至少一种能用的 C 编译器。并且它的语法和函数库在不同的平台上都是统一的，这个特性对开发者来说很有吸引力。

2）用 C 语言写的程序执行速度很快。

3）C 语言是所有版本的 Unix 上的系统语言。

Linux 上可用的 C 编译器是 GNU C 编译器，它建立在自由软件基金会的编程许可证的基础上，因此可以自由发布。你能在 Linux 的发行光盘上找到它。

本讲主要讲解 C 语言编程和编译的基本方法和常用工具。

10.2 C 语言编程工具教学目标（4 学时）

1. 教学目标

最终目标：掌握 C 程序设计的基本方法和常用工具的使用。

促成目标：

1）掌握 C 语言多文档编程编译技巧和方法；

2）掌握归档工具的使用；

3）熟悉 make 文件的编写和编译方法；
4）掌握 C 语言调试工具的使用。
2. 工作任务
1）编写一个多文档的 C 语言程序；
2）用归档工具实现归档和提取；
3）编写 make 文件，实现用 make 编译 C 语言程序；
4）用调试工具调式 C 语言程序。
3. 活动设计
编写一个加法程序，分别用参数形式输入和文件输入形式实现。

10.3　C 语言编程工具　实验指导

10.3.1　C 编译器

C 编译器是 Unix 的标准组件。在 Linux 分发版本中包含 GNU C（gcc）和 GNU C ++（g ++）两个工具。

编译程序语法：gcc – cv [-o filename] [-pg] {filename} *
-c：选项生成的是目标模块；
-v：选项生成详细输出，可以看到 gcc 所做的工作；
-pg：选项生成分析数据，供 GNU 分析器 gprof 使用；
-o：选项指定所生成的可执行文件名。

10.3.2　单模块程序

编写一个程序 test. c：
```
#include <stdio. h>
main( )
{
   printf("Hello ! \n");
}
```
编译该程序：
```
$ gcc test. c
$ ls
a. out   test. c
```
默认生成可执行文件 a. out，执行该文件：
```
$ ./a. out
Hello!
```
指定生成可执行文件名的编译：
```
$ gcc  -o test test. c
$ ls
```

```
test test.c
```

执行该程序：

```
$./test
Hello!
```

10.3.3　多模块程序

1. 创建可重用函数

将程序中的函数分离出来分别编译，可用于不同的程序，有利于提高编程效率，也使得程序结构更加清晰，便于管理。

例如，有一个加法程序，将加法函数分离出来编译如下：

```
$ cat add.h
/*add.h*/
int add(int a,int b);
$ cat add.c
#include  "add.h"
add(a,b)
{
      return  a+b;
}
$ cat main.c
#include  "add.h"
main()
{
   int a=10,b=20;
   int c=add(a,b);
   printf("%d+%d=%d",a,b,c);
}
```

2. 编译模块

```
$ gcc  -o mainadd main.c add.c
```

执行程序 mainadd：

```
$./mainadd
10+20=30
```

也可以先编译成模块,再合成：

```
$ gcc  -c add.c main.c
$ ls
add..c add.h add.o main.c main.o
$ gcc  -o mainadd main.o add.o
```

10.3.4　归档模块 ar

一个中等大小的C程序项目要使用几百个目标模块。在命令中指定这么多模块不但单

调，而且容易出错，所以建议使用 GNU 的归档模块 ar 来组织和分组目标模块。归档使用程序也叫“库管理器”，它能执行以下任务：

1）创建一个扩展名为“. a”的归档格式文件；

2）在归档文件中添加、删除、替换和追加任意类型的文件；

3）获得归档的内容表。

1. 创建或添加文件并建立索引

ar rs archiveName{filename} +

归档在添加第一个文件时自动创建，若文件已存在则替换文件，归档文件 archiveName 应以“. a”作为扩展名。例如：

```
$ ar rs testar.a add.o add.h
```

2. 查看归档文件中的内容

ar t archiveName

例如：

```
$ ar t testar.a
add.o
add.h
```

3. 追加文件

ar q archiveName{filename} +

在 archiveName 文件中追加指定的文件，不管归档文件中是否存在这些文件。例如：

```
$ ar q testar.a main.o test.o
$ ar t testar.a
add.o
add.h
main.o
test.o
```

4. 删除文件

ar d archiveName {filename} +

从归档文件中删除列出的文件，例如：

```
$ ar d testar.a test.o
$ ar t testar.a
add.o
add.h
main.o
```

5. 抽取文件

ar x archiveName {filename} +

从归档文件中复制列出的文件到当前目录，例如：

```
$ ar x testar.a add.o main.o
```

10. 3. 5 编译管理工具 make

将几个独立的模块连接到一个可执行文件，不需要什么工具就可以解决，但如果该项目

很大，有几十个甚至几百个模块，要记住所有的头文件、源代码、目标模块和可执行文件之间的关系是多么复杂。当进行了修改，就要把所有的文件重新编译，不仅浪费时间，而且浪费资源。避免这个问题的有效方法就是使用 GNU 的 make 工具，通过 makefile 文件，列出每个可执行文件的的所有依赖关系。make 程序用法如下：

make[-f makefile]

-f 选项允许用户指定自己的 make 文件名，如果不指定，make 将依次查找 GNUmakefile、makefile 和 Makefile。

Makefile 文件的最简形式是：

```
TargerList：dependencyList
            commandList
```

注意：commandList 前必须有一个空制表位，否则要出错。

例如，将前面的加法程序用 makefile 文件重写一遍如下：

```
$ cat  makefile
mainadd:main. o  add. o
        gcc  main. o  add. o  -o  mainadd
main. o:main. c  add. h
        gcc  -c  main. c
add. o:  add. c  add. h
        gcc  -c  add. c
$ make                       ;编译生成程序
```

由于 make 程序还能推断目标模块名字和它的源代码文件通常是相关的，使用这个信息推断标准的依赖性。如 main. o 依赖于 main. c，add. o 依赖于 add. c，因此在依赖性列表中可以省略这个信息，使 makefile 更简洁：

```
$ cat  makefile
mainadd:main. o  add. o
        gcc  main. o  add. o  -o  mainadd
main. o:  add. h
add. o:  add. h
```

其效果与原来的 makefile 相同。

10. 3. 6　make 工具与 ar 归档工具联合应用

首先要建立有索引的归档文件，例如：

```
$ ar  rs  testar. a  add. c  add. h  main. c
```

然后编辑 makefile 文件：

```
$ cat  makefile
mainadd:testar. a(main. o)  testar. a(add. o)
            gcc  testar. a  -o mainadd
testar. a(add. o):testar. a(add. h)
testar. a(main. o):testar. a(add. h)
```

```
$ make                ;编译生成程序
```

10.3.7 调试器 gdb

GNU 调试器 gdb 以符号方式调试程序，它是一个非常方便的实用程序。gdb 的功能如下：

1）运行和列出程序；

2）设置断点；

3）检查变量值；

4）追踪执行。

1. 准备调试程序

为了要用 gdb 工具调试程序，在编译时需要加-g 选项，并且编译程序文档不能放在归档文件中。例如：

```
$ gcc -g -c add.c main.c
$ gcc add.o main.o -o mainadd
```

2. 进入调试器

```
$ gdb mainadd                 ;调试 mainadd 程序
(gdb)help                     ;获取帮助信息
(gdb)run                      ;运行程序
(gdb)list                     ;列出程序
1         #include"add.h"
2         main()
3         {
4         Int a=10,b=20;
5         printf("%d+%d=%d\n",a,b,add(a,b));
6         }
(gdb)list add.c:1,10          ;列出 add.c 程序的 1~10 行
1         #include "add.h"
2         add(a,b)
3         {
4         return a+b;
5         }
(gdb)break add                ;在调用 add 函数处设置断点
(gdb)run                      ;从头开始运行程序
Breakpoint 1, add(a=10,b=20) at add.c:4
4         return a+b;
(gdb)step                     ;单步执行,也可以执行多步,如 step 5
(gdb)break 5                  ;在行号 5 处设置断点
(gdb)run                      ;重新运行程序
Breakpoint 1,main() at main.c:5
```

```
5           printf("%d + %d = %d\n",a,b,add(a,b));
(gdb)print  a                 ;检查变量 a 的值
$ 1 =10
(gdb)step                     ;单步执行
add(a =10,b =20)  at  add.c:4
(gdb)print  b                 ;检查变量 b 的值
$ 2 =20
(gdb)quit                     ;退出调试程序
```

附录

常用命令索引及详解

1. 文件处理指令

(1) 列表指令 ls

指令英文原义：list

指令所在路径：/bin/ls

执行权限：All User

语法：ls 选项[-alFR][文件或目录]

```
-a    ;显示所有文件,包括隐藏文件
-l    ;使用 long format(详细列表)
-F    ;附加文件类别,符号在文件名最后
-R    ;显示所有子目录
```

功能描述：显示目录文件。

范例：

$ ls-a ;显示所有文件,包括隐藏文件

```
[cwb@fanhs ~]$ ls -a
.       add.h            .bash_logout   .emacs  mainadd  makefile
..      add.o            .bash_profile  .gtkrc  main.c   testar.a
add.c   .bash_history    .bashrc        .kde    main.o   .zshrc
```

$ ls-l ;详细列表

```
[cwb@fanhs ~]$ ls -l
总用量 68
-rw-rw-r--  1 cwb cwb   44  9月 23 15:21 add.c
-rw-rw-r--  1 cwb cwb   22  9月 23 15:21 add.h
-rw-rw-r--  1 cwb cwb 1788  9月 23 15:22 add.o
-rwxrwxr-x  1 cwb cwb 6140  9月 23 15:22 mainadd
-rw-rw-r--  1 cwb cwb   82  9月 23 15:21 main.c
-rw-rw-r--  1 cwb cwb 2156  9月 23 15:22 main.o
-rw-rw-r--  1 cwb cwb  173  9月 23 15:11 makefile
-rw-rw-r--  1 cwb cwb 2162  9月 23 14:49 testar.a
```

(2) 建立目录、删除目录指令 mkdir rmdir

指令英文原义：make directory，remove directory

指令所在路径：/bin/mkdir，/bin/rmdir

执行权限：All User

语法：mkdir [-pm]目录名 1　目录名 2　...

　-p　　;若父目录不存在,此选项可自动建立

　-m　　;为新建目录设置存取权限,用 8 进制表示

语法:rmdir [-p]　目录名

　-p　　递归删除目录,当子目录被删除后,父目录为空时也一同删除,如果是非空目录,则保留下来

范例：

```
$ mkdir  a  b  c  d          ;在当前目录下建立 a、b、c、d 目录
$ mkdir  -p  a1/a2/a3        ;建立 a1/a2/a3 目录,a1 不存在
$ mkdir  -m 765  test        ;建立 test 目录,存取权限为 765
$ rmdir  a  b  c             ;删除空目录 a、b、c,若目录不空则不能删除
```

注：非空目录可用 rm-r 命令删除。

(3) 改变目录指令 cd

指令英文原义：change directory

指令所在路径：Shell 内部指令

执行权限：All User

语法：cd [目录]

范例：

```
$ cd  ..           ;返回上层目录
$ cd  .            ;在原目录
$ cd ~             ;回到 User 的 Home directory(用户根目录)
$ cd  /            ;切换到系统根目录
$ cd  ../user      ;切换到上一层目录的 Usr 目录
```

(4) 路径显示指令 pwd

指令英文原义：print work dirctory

指令所在路径：/bin/pwd

执行权限：All User

语法：pwd

功能描述：显示目前所在的工作目录。

范例：

由于 Linux 系统的树状目录结构颇为庞大，若没有使用 Shell 环境变量指定在 Shell prompt 显示目前的工作目录，而不断地使用 cd 切换目录，可能会忘记目前所在的目录是哪里。例如，切换到/usr/local/bin，一般 Shell prompt 只会显示最后的目录名称/bin，所以会对所在目录产生混淆，因此随时使用 pwd 指令，可以让您了解目前的工作目录。

```
[cwb@fanhs bin]$ cd /usr/local/bin
[cwb@fanhs bin]$ pwd
/usr/local/bin
```

（5）复制、删除、移动指令 cp、rm、mv

指令英文原义：copy，remove，move

指令所在路径：/bin/cp，/bin/rm，/bin/mv

执行权限：All User

语法：cp-afpx［源文件或目录］［目的文件或目录］

-a ;archive 的意思,也就是复制所有目录并包含子目录

-f ;强制复制文件

-p ;保留原文件日期

-x ;指定文件复制后属于目的磁盘的文件系统,如 ext2-> fat32

语法：rm -irf［文件或目录］

-i ;互动模式,删除前再作一次确认

-f ;强制删除

-r ;删除目录及子目录

语法：mv-if［源文件或目录］［目的文件或目录］

功能描述：复制、删除、移动文件。

范例：

```
$ cp file1  file2  dir        ;将文件 file1、file2 复制到 dir 目录
$ cp  -ax  dir1  dir2         ;将 dir1 下所有的目录包括子目录全部复制到 dir2
```

在 Linux 上常会发生跨文件系统的复制，例如：Linux ext2 file system 的文件复制到 Windows FAT32 file system 上，所以指定-x 让文件随所在目录的文件系统作改变。

```
$ rm  -i file          ;删除前询问是否删除
$ rm  -rf dir          ;强制删除整个目录
$ mv file1  file2      ;将文件 file1 移动到 file2,其实就是更名操作。若 file2 是目录,
                        则是将 file1 移动到 file2 目录下
```

最后需要指出，cp、rm、mv 是任务用户都可以使用的指令，但如果文件权限不对，则会出现 Permission Denied 的错误信息，显示权限不足，不能复制、删除和移动文件。

（6）文件链接指令 ln

指令英文原义：link

指令所在路径：/bin/ln

执行权限：All User

语法：ln-s［源文件］［目标文件］

-s symbolic link （符号链接）

功能描述：产生链接文件。

范例：

当需要在不同的目录用到相同的文件时，不需要在每一个需要的目录下放一个必需的相同文件，只要在某个固定的目录上放上该文件，然后在其他目录用 ln 这个指令链接它即可，不必重复地占用硬盘空间。例如：

```
$ ln  -s  /bin/less  /usr/local/bin/less    ; 将/bin/less 连接到/usr/local/bin/less
```

这样，在/usr/local/bin 下也会产生一个 less 文件名，因为使用了符号链接（symbolic

link），所以该文件只占用几个字节。用 ls-l 查看，可以看到 link 的路径，例如：less- > /bin/less。

假如没有使用-s 选项，直接用 ln 链接，这样的链接称为硬链接（hard link），其功能是为同一个文件产生多个链接，但链接文件大小不变，仍然需要占用空间。

两种链接文件都能使得每个链接的文件同时变化，即无论编辑或改变哪个链接文件的内容，所有的链接文件都会发生同样的变化。所不同的是：

1）符号链接的文件只有几个字节大小，并不会随着主文件的变化而改变。例如，主文件从几 K 字节变化到几个 G，链接文件不会随之变大，仍然是几个字节，而硬链接的文件则要同时变大或变小。

2）符号链接的文件依赖于主文件，若主文件被删除，则链接文件就会出错。当然，链接文件可以删除，不影响主文件。而硬链接的文件却不依赖于主文件，虽然同时变化，但任何一个链接文件删除都不会影响其他文件。

2. 文件权限设置

（1）改变文件权限指令 chmod

指令英文原义：Change file or directory's mode

指令所在路径：/bin/chmod

执行权限：All User

语法：chmod　-R　[mode = 4,2,1 or{a,u,g,o}{+,-,=}{r,w,x,s,t}][文件或目录]

功能描述：改变文件或目录的权限。

范例：

将文件的权限分成 Readable（r）、Writable（w）、eXecutable（x）、Set User ID Setuid（s）、Set Group ID Setgid（s）、Sticky bit（t），将文件的用户分成 All（a）、User（u）、Group（g）、Other（o）四种。可以使用 chmod 加上这些代号规范文件权限。

```
$ chmod  a+r   file      ;让所有用户对 file 文件有读的权限
$ chmod  go+x  file      ;给同组用户和其他用户加上可执行权限
$ chmod  u=rwx  file     ;设置 file 文件所有者的权限为可读、可写、可执行
$ chmod  a+t  /tmp       ;将/tmp 目录的权限加上 t,这样任何人在/tmp 目录所写的文
                          件就不会被其他用户删除(除文件所有者和 root 用户)
```

文件权限也可以用数字来表示，分成 4 段，每段用一个 8 进制数来表示：

特权（可选）	用户权限	同组用户权限	其他用户权限
0~7	0~7	0~7	0~7

每段用 2 进制表示 3 个值，0 表示无权，1 表示有权限，最后合成 8 进制数：

Readable（r）	Writable（w）	eXecutable（x）	最后合成
0~1	0~1	0~1	0~7

特权段（可选）：

Setuid（s）	Setgid（s）	Sticky bit（t）	最后合成
0~1	0~1	0~1	0~7

```
$ chmod  700  file   ;用户有所有的权限,同组用户和其他用户无权限
$ chmod  4755  file  ;用户有所有的权限,同组用户和其他用户有可读、可执行权限,并
```

且因为设置了 Setuid,所以不管执行者是谁,都是以文件所有者的身份在执行。Setuid 权限的设置通常发生在需要 root 权限才能执行的情况下,这样,就不需要 root 亲自动手,其他用户也能代替 root 执行该文件的操作。

$ chmod -R 655 mydir/ ;将 mydir 目录下所有的文件权限设置为 655

(2) 改变文件所有者指令 chown

指令英文原义：Change file or directory's Owner

指令所在路径：/bin/chown

执行权限：All User

语法：chown -R[用户.[同组用户]][文件或目录]

功能描述：改变文件的所有者。

范例：

chown user1 file1 ;将 file1 的所有者改为 user1

chown -R user1. group1 mydir/ ;将 mydir 目录下的文件的所有者改为 user1，组改为 group1

注意：改变所有者一般需要 root 来执行，因为其他用户没有权限。

(3) 改变文件组指令 chgrp

指令英文原义：Change file or directory's Group

指令所在路径：/bin/chgrp

执行权限：All User

语法：chgrp -R [文件或目录]

功能描述：改变文件的所属组

范例：

chgrp group1 file1 ;将 file1 文件的组属性改为 group1

chgrp -R group1 mydir/ ;将 mydir 目录下所有的文件组属性改为 group1

3. 文件显示、排序、统计

(1) 文件显示指令 cat、more、less、head、tail、uniq

语法：cat ［选项］ 文件名

功能描述：显示全部文件，若内容超出屏幕一页以上也不会停止显示，直到显示完成。适合显示小文件（内容小于一页）。

cat 命令连接文件并打印到标准输出设备上。cat 经常用来显示文件的内容，就像 DOS 系统的 type 命令。

cat 命令还有一项功能，用它可以连接两个或多个文件。例如：

$ cat f1 f2 > f3 ;将 f1、f2 合并到 f3 文件，"＞"是重定向符

还可以用 cat 命令创建一个新文件，通过标准输入设备键入文件内容，按 ctrl + D 键结束：

$ cat > 文件名

在标准终端输入文件内容

Ctrl + D 结束

cat 选项功能如下：

-b　　;从 1 开始对所有非空输出行进行编号
-n　　;从 1 开始对所有输出行进行编号
-s　　;将多个相邻的空行合并为一个空行

例如：

```
[cwb@fanhs ~]$ cat main.c
#include "add.h"


main()
{
  int a=10,b=20;
  printf("%d+%d=%d\n",a,b,add(a,b));
}
[cwb@fanhs ~]$ cat -n main.c
     1  #include "add.h"
     2
     3
     4  main()
     5  {
     6    int a=10,b=20;
     7    printf("%d+%d=%d\n",a,b,add(a,b));
     8  }
```

```
[cwb@fanhs ~]$ cat -n -s main.c
     1  #include "add.h"
     2
     3  main()
     4  {
     5    int a=10,b=20;
     6    printf("%d+%d=%d\n",a,b,add(a,b));
     7  }
```

语法：more　[选项]文件名

选项功能：

-num　　;指定一个整数,表示一个屏幕显示的行数
-d　　;在每屏底部显示“Press space to continue‘q’to quit”,并在用户输入非功能键时显示“Press‘b’for instructions”信息
-p　　;不滚屏,在显示下一屏之前先清屏
-s　　;文件中连续的空白行合并成一行输出

以上选项可以联合使用。

功能描述：显示文件，满一页屏幕时停止等待，并显示剩余比例值，按回车下移一行，

按空格下移一页，但不能往回移动。

语法：less ［选项］ 文件名

功能描述：less 是 more 的改进版，显示文件，满一页屏幕时等待，可以用光标键上移或下移屏幕，也可以用 PageUp 和 PageDown 键翻页，按 q 键才能退出显示。

语法：head[-n] 文件名

功能描述：显示文件的前 n 行，不加选项-n 时显示前 10 行

语法：tail[-n] 文件名

功能描述：显示文件的最后 n 行，不加选项-n 时显示最后 10 行

语法：uniq 文件名

功能描述：显示文件中连续的不重复的行，一般需要先将文件排序，才能保证显示所有的不重复的行。

（2）文件排序指令 sort

语法：sort ［选项］ 文件列表

选项功能：

-m ;如果文件列表中文件已排序,则对这些文件统一进行合并

-r ;逆向排序

-o ;将排序输出的内容放到该指定文件中,若文件不存在则创建一新文件

例如：

```
[cwb@fanhs ~]$ cat a.t
a
c
e
[cwb@fanhs ~]$ cat b.t
b
d
f
[cwb@fanhs ~]$ sort -m a.t b.t -o c.t
[cwb@fanhs ~]$ cat c.t
a
b
c
d
e
f
[cwb@fanhs ~]$ sort -r c.t
f
e
d
c
b
a
```

（3）文件内容统计指令 wc

语法：wc ［选项］ 文件名

选项功能：

- -c ;统计字节数
- -l ;统计行数
- -w ;统计字数

例如：

```
[cwb@fanhs ~]$ cat -n main.c
     1  #include "add.h"
     2  main()
     3  {
     4    int a=10,b=20;
     5    printf("%d+%d=%d\n",a,b,add(a,b));
     6  }
[cwb@fanhs ~]$ wc -l main.c
6 main.c
```

4. 文件及文件内容查找

（1）文件查找指令 find

语法：find ［路径］［表达式］

选项功能：

- -group 组名 ;找出属于指定组的所有文件
- -user 用户名 ;找出属于指定用户的所有文件
- -print ;在屏幕输出找到的文件
- -fprint 文件名 ;将找到的文件放入该指定文件中
- -exec 指令{}\; ;对符合条件的文件执行指定的指令
- -ok 指令{}\; ;条件与 exec 相同,但找到的每个文件需经过用户确认方能执行该指令

功能描述：在目录中查找满足条件的文件，并执行指定的操作。

find 命令从左向右分析各个参数，然后依次搜索目录。如果没有设置路径，find 就搜索当前路径。

表达式由选项、测试和操作 3 部分组成，分别由运算符分开。

例如：

```
[root@fanhs cwb]# ls -l
总用量 84
-rw-rw-r--  1 cwb cwb   17  9月 24 07:27 abc.find
-rw-rw-r--  1 cwb cwb   44  9月 23 15:21 add.c
-rw-rw-r--  1 cwb cwb   22  9月 23 15:21 add.h
-rw-rw-r--  1 abc abc 1788  9月 23 15:22 add.o
-rw-rw-r--  1 cwb cwb  240  9月 24 07:23 cwb.find
```

```
-rwxrwxr-x  1 cwb cwb 6140  9月 23 15:22 mainadd
-rw-rw-r--  1 cwb cwb   82  9月 24 05:16 main.c
-rw-rw-r--  1 abc abc 2156  9月 23 15:22 main.o
-rw-rw-r--  1 cwb cwb  173  9月 23 15:11 makefile
-rw-rw-r--  1 cwb cwb 2162  9月 23 14:49 testar.a
[root@fanhs cwb]# find ./ -user abc -fprint abc.find
[root@fanhs cwb]# cat abc.find
./main.o
./add.o
[root@fanhs cwb]# find ./ -user abc -print -exec rm {} \;
./main.o
./add.o
[root@fanhs cwb]# find ./ -user abc -print
[root@fanhs cwb]#
```

（2）文件内容查找指令 grep

语法：grep ［选项］ 文件名

功能描述：用来在文本文件中查找指定的模式的单词或短语，并在标准输出上显示包括给定字符串模式的所有行，在指定文件中搜索特定模式及特定主题等方面用途很大。

如果没有指定文件，就从标准输入中读取。在很多情况下从管道的上线读取数据。在正常情况下，每个匹配的行被显示到标准输出上。如果要搜索的文件不止一个，则在每一行输出之前加上文件名。

选项功能：

- -b ;在输出的每一行前面显示包含匹配字符串的行在文件中的位置,用字节偏移量表示
- -c ;只显示文件中包含匹配字符串行的总数
- -i ;匹配比较时不区分大小写
- -r ;以递归方式查询目录下所有的子目录中的文件
- -n ;在输出包含匹配模式行之前加上该行的行号

例如：

```
[root@fanhs cwb]# cat main.c
#include "add.h"
main()
{
  int a=10,b=20;
  printf("%d+%d=%d\n",a,b,add(a,b));
}
[root@fanhs cwb]# grep -n add main.c
1:#include "add.h"
5:  printf("%d+%d=%d\n",a,b,add(a,b));
[root@fanhs cwb]# mkdir -p a/a1/a2
```

```
[root@fanhs cwb]# cp main.c a/a1/a2
[root@fanhs cwb]# grep -r -n add a
a/a1/a2/main.c:1:#include "add.h"
a/a1/a2/main.c:5:  printf("%d+%d=%d\n",a,b,add(a,b));
[root@fanhs cwb]# grep -r -c add a
a/a1/a2/main.c:2
[root@fanhs cwb]#
```

5. 系统帮助指令

（1）联机帮助手册指令 man

语法：man ［选项］ 命令

功能描述：用于显示某一命令的联机帮助手册页。

选项功能：

-f ;只显示命令的功能,而不显示其中详细的说明文件

-w ;不显示手册页,只显示文件所在位置

-a ;显示所有手册页(默认)

```
[root@fanhs cwb]# man -f  cpio
cpio                 (1)  - copy files to and from archives
[root@fanhs cwb]# man -w cpio
/var/cache/man/cat1/cpio.1.bz2 (<-- /usr/share/man/man1/cpio.1.g
```

（2）帮助信息指令 info

语法：info 命令

功能描述：显示指定命令的相关信息。

6. 文件压缩指令

（1）单个文件压缩、解压缩指令 gzip、gunzip

语法：gzip ［选项］ 压缩文件名/解压缩文件名

功能描述：用于对单个文件进行压缩解压缩，被压缩的文件扩展名是 . gz。

选项功能：

-c ;将输出写到标准输出上,并保留原有文件

-d ;将被压缩文件进行解压

-r ;递归地查找指定目录并压缩或解压缩其中的所有文件

-t ;检查压缩文件的完整性

例如：

$ gzip main. c ;压缩 main. c 文件

$ gzip -d main. c. gz ;解压缩 main. c. gz,还原成 main. c

$ gzip-r a ;递归压缩 a 目录包括子目录中的所有文件

$ gzip -r-d a ;递归解压缩 a 目录中所有文件

$ gunzip-r a ;采用 gunzip 命令解压,效果同上

（2）目录打包压缩指令 tar

语法：tar ［选项］ 目录名

功能描述：对目录进行打包和压缩，经常用它作系统备份。

选项功能：

- -f ;指定打包压缩存档文件名
- -c ;创建一个新文档
- -r ;将文件附加到已存在的文档后面
- -u ;仅添加比文档文件中更新的文件
- -x ;从文档中恢复被备份的文件
- -t ;不解压或解包显示备份文档中文件名
- -z ;在打包同时进行压缩或解压缩

例如：

```
[root@fanhs cwb]# ls
abc.find  add.c  add.h  mainadd  main.c  makefile  testar.
[root@fanhs cwb]# mkdir a
[root@fanhs cwb]# cp a* a
cp: 略过目录 'a'
[root@fanhs cwb]# ls a
abc.find  add.c  add.h
[root@fanhs cwb]# tar cvf a.tar a
a/
a/add.h
a/abc.find
a/add.c
[root@fanhs cwb]# cp m* a
[root@fanhs cwb]# tar uvf a.tar a
a/
a/makefile
a/main.c
a/mainadd
[root@fanhs cwb]# tar rvf a.tar testar.a
testar.a
```

```
[root@fanhs cwb]# tar tvf a.tar
drwxr-xr-x root/root         0 2010-09-24 10:20:04 a/
-rw-r--r-- root/root        22 2010-09-24 10:20:04 a/add.h
-rw-r--r-- root/root        17 2010-09-24 10:20:04 a/abc.fi
-rw-r--r-- root/root        44 2010-09-24 10:20:04 a/add.c
drwxr-xr-x root/root         0 2010-09-24 10:20:54 a/
-rw-r--r-- root/root       173 2010-09-24 10:20:54 a/makefi
-rw-r--r-- root/root        82 2010-09-24 10:20:54 a/main.c
-rwxr-xr-x root/root      6140 2010-09-24 10:20:54 a/mainad
-rw-rw-r-- cwb/cwb        2162 2010-09-24 08:13:42 testar.a
[root@fanhs cwb]#
```

7. 网络指令

(1) 显示设置主机名指令 hostname

语法：hostname ［选项］ ［主机名］

功能描述：当没有任何参数时，就显示系统的主机名。设置一个参数时，就将该参数作为系统新的主机名。

例如：

$ hostname　　　　　　;显示系统主机名

hostname　abcd　　;将系统主机名设置为 abcd,当然,操作者必须为 root 用户

(2) 网络连通性测试指令 ping

语法：ping ［选项］ 主机名(或 IP 地址)

功能描述：测试本主机和目标主机的连通性。Ping 命令使用 ICMP 协议，向网络主机发送 ECHO_ REQUEST 数据包，希望获得目的主机的 ICMP ECHO_ RESPONSE 应答数据包，以判断和网络主机之间的连通情况。

选项功能：

-c　发送数　　　　;发送指定数量的 ECHO_REQUEST 数据包

-s　包长度　　　　;指定发送数据包的大小,默认为 56 字节,加上 8 个字节的 ICMP 头信息,实际发送 ICMP 数据包为 64 个字节

-t　数字　　　　　;设置数据包的存活时间 TTL

数据包都有 TTL，每过一个路由器，TTL 将会被减 1，如果 TTL 等于 0，路由器将会抛弃这个数据包。

-i　　　　　　　　;设置两次信息之间的间隔时间,若不加此项,间隔默认为 1s

例如：

$ ping -c 5 -i 2 -s 10 192.168.1.100　;向 192.168.1.100 主机发送 5 个测试数据包,时间间隔为 2s,数据包长度为 10。

(3) 网络设备检查设置指令 ifconfig

语法：ifconfig　网卡号 ［选项］［IP 地址］

功能描述：用该指令显示网络设备的状态，配置网络设备的临时参数。

选项功能：

-up　　　　　;激活网卡

-down　　　　;关闭网卡

新指令：

ifup　网卡号　　　;激活网卡

ifdown　网卡号　　;关闭网卡

例如：

$ ifconfig　eth0　　　　　　　　;检查网卡 eth0 的配置情况

$ ifconfig　eth0　192.168.1.200/24　;设置网卡的临时 IP 地址

$ ifconfig　eth0　down　　　　　;关闭网卡 eth0

$ ifdown　eth0　　　　　　　　　;效果同上

```
$ ifconfig  eth0  up                ;激活网卡
$ ifup   eth0                       ;效果同上
```

(4) 路由检查和设置指令 route

语法：route ［add/del］［-net IP 地址/-host IP 地址/default］［gw］［IP 地址］

功能描述：显示系统路由设置情况，设置默认路由、指定路由（到网络或主机）。

选项功能：

```
add                ;增加路由设置
del                ;删除路由设置
-net   IP 地址      ;指定到达网络的路由地址
-host  IP 地址      ;指定到达主机的路由地址
default            ;指定默认路由
gw                 ;指定网关
```

例如：

```
# route                                            ;显示当前路由设置
# route  add  default gw 192.168.1.254             ;设置默认路由
# route  add-net 104.101.0.0/16 gw 192.168.1.200   ;设置指定到网络的路由
# route add-host  105.110.1.1  gw  192.168.1.150   ;设置指定到主机的路由
# route del  -net 104.101.0.0/16 gw 192.168.1.200  ;删除路由
```

(5) 网络启动、重启和关闭指令 network

语法：service network start/restart/stop

功能描述：网络启动、重启和关闭。

8. 系统管理指令

(1) 进程显示指令 ps

语法：ps ［aux］

功能描述：显示系统的进程信息。

ps 指令输出符号含义：

UID：　　启动进程的用户名；

PID：　　进程号；

PPID：　　父进程号；

TTY：　　启动进程的终端号；

STAT：　　进程状态；

START：　　进程开始时间；

TIME：　　进程已经运行时间；

COMMAND：进程的命令名。

例如：

```
# ps  aux                    ;显示所有进程
```

(2) 终止进程指令 kill

语法：kill ［-9］ 进程号

功能描述：终止一个进程的运行。

选项功能：

-9　　;强制终止进程的运行

例如：

$ kill　-9　12345　　;终止 12345 号进程的执行

(3) 进程暂停执行指令 sleep

语法：sleep　时间(秒)

功能描述：使进程暂停执行指定的时间（秒）。

例如：

$ sleep　20　　;暂停执行进程 20s

(4) 清除屏幕指令 clear

语法：clear

功能描述：清除屏幕内容，使提示符显示在左上角。

(5) 显示登录用户指令 who

语法：who　[选项]

功能描述：列出所有正在使用系统的用户，whoami 命令显示当前用户信息。

选项功能：

-H　　;打印各列标题和内容

-m　　;仅显示各列内容

(6) 日期时间显示和设置指令 date

语法：date　[选项]　日期时间

功能描述：显示系统日期时间，并可设置新的日期和时间。

功能选项：

MMDDhhmm[cc][yy][.ss]　　;设置日期和时间

-s　[CC]YYMMDD　　;设置日期

例如：

date　101520302010　　;设置日期时间为 2010 年 10 月 15 日 20 时 30 分

date　-s　20101015　　;设置日期为 2010 年 10 月 15 日

(7) 日历显示指令 cal

语法：cal　[[月]　年]

功能描述：显示指定年、月的日历。

例如：

$ cal　2010　　;显示 2010 年的日历

$ cal　10　2010　　;显示 2010 年 10 月的日历

(8) 系统关机指令 halt　shutdown　init

语法：halt

功能描述：立即关机，不通知用户。

语法：shutdown　[选项]

功能描述：可定时关机，并在关机前通知用户，也可用于重启系统。

选项功能：

-h ;关机参数,可在后跟关机延迟时间(分钟),可提示信息,并在各终端显示以通知各用户注意

-r ;重启参数,其他同上

例如:

shutdown -h 3 The machine will be shutdown after 3 minutes…
;系统在 3min 后关机,并显示后跟信息

语法:init [0-6]

功能描述:用该指令根据不同的参数,可以实现多个系统状态转换

选项功能:

0 ;立即关机
1 ;进入单用户状态
2 - 3 ;进入字符多用户状态
4 ;未用
5 ;图形多用户状态
6 ;重启系统

例如:

init 0 ;立即关机

9. 文件系统管理指令

(1) 文件系统挂接指令 mount/umount

语法:mount [-t 文件类型] 设备名 挂接目录

功能描述:将该文件系统设备挂接到指定目录。

常用设备名:

/dev/cdrom ;光盘驱动器
/dev/fd0 ;软盘驱动器
/dev/sda1 ;第一个 U 盘

例如:

mount /dev/cdrom /mnt ;将光盘驱动挂接到/mnt 目录
mount -t vfat /dev/sda1 /mnt ;将 U 盘 fat 格式文件系统挂接到/mnt 目录

语法:umount 设备名/挂接目录

功能描述:将文件系统设备卸载,可指明需要卸载的设备名或挂接目录。

例如:

umount /dev/cdrom

或

umount /mnt ;卸载挂接在/mnt 目录的光盘驱动。

(2) 文件系统挂接情况显示 df

语法:df [-m]

功能描述:显示文件系统挂接情况,可在文件系统挂接或卸载后检查用。

例如:

```
[root@fanhs ~]# df
Filesystem           1K-块      已用      可用 已用% 挂载点
/dev/sda1          4648864   2910920   1501792  66% /
none                130052         0    130052   0% /dev/shm
[root@fanhs ~]# df -m
Filesystem           1M-块      已用      可用 已用% 挂载点
/dev/sda1             4540      2843      1467  66% /
none                   128         0       128   0% /dev/shm
[root@fanhs ~]#
```

(3) 文件目录空间占用显示指令 du

语法：du 目录[-b|k|m][-s]

功能描述：显示指定目录的空间占用量。

参数功能：

-b|k|m ;空间占用量用 B 或 KB 或 MB 显示

-s ;只显示目录总占用量,不显示子目录占用量

例如：

```
[root@fanhs ~]# du /dev/ -b
60      /dev/mapper
160     /dev/snd
120     /dev/input
60      /dev/net
40      /dev/shm
0       /dev/pts
169847  /dev/
[root@fanhs ~]# du /dev  -bs
169847  /dev
```

10. 用户管理指令

(1) 添加组指令 groupadd

语法：groupadd ［参数］ 组名

功能描述：建立新组。

参数功能：

-g GID ;指定新组的 GID,默认值是现有最大的 GID 加 1

例如：

groupadd grp1 ;建立新组 grp1,可以在/etc/group 看到

groupadd -g 1000 grp1000 ;建立新组 grp1000,指定 GID

(2) 添加用户指令 useradd

语法：useradd ［参数］ 用户名

功能描述：建立新用户。

参数功能：

-c comment ;注释行,可填写名称、地址、电话等信息

```
-d   dir                 ;设置指定目录,默认为/home/用户名
-e   YYYY-MM-DD          ;设置用户有效日期,此后则不能使用
-g   group               ;指定组编号 GID 或组名
-s   shell               ;指定用户登录后启用的 shell,默认是/bash shell
-u   UID                 ;指定用户编号,默认是当前最大的 UID 加 1
```

例如:

```
# useradd  user1                     ;添加用户 user1
# useradd  -g  group1  user2         ;添加用户 user2,指定组名 group1
# useradd  -d  /usr/user3  -e  2010-12-31  user3
                                     ;添加用户,指定目录、有效期
# useradd  -s  /bin/false  user4     ;添加用户并指定无效 shell,这样该用户不能登
                                      录,但具有其他用户功能
```

(3) 删除用户指令 userdel

语法:userdel [-r] 用户名

功能描述:删除用户账户,必须加-r 参数,否则不能删除用户目录,并给以后添加新的相同用户增加麻烦。

例如:

```
# userdel  -r   user1                ;删除用户 user1,同时删除用户目录
```

(4) 修改用户属性指令 usermod

语法:usermod [参数] 用户名

功能描述:修改已有用户的属性,参数功能同 useradd。

例如:

```
# usermod  -c  这是 user1 的注释  user1   ;修改 user1 的注释
# usermod  -e  2011-1-1 user4            ;将 user4 的有效期改变
```

(5) 修改用户口令指令 passwd

语法:passwd [-d][用户名]

功能描述:修改指定用户或当前登录用户的口令,修改指定用户需超级用户权限。

其他功能:用参数-l |-u 可用此命令停止和恢复用户。

例如:

```
# passwd  user1          ;修改 user1 的口令(需输入两次)
 $ passwd                ;修改登录用户自身的口令
# passwd  -d  user1      ;删除 user1 用户的口令
# passwd  -l  user1      ;暂停用户 user1
# passwd  -u  user1      ;恢复用户 user1
```

注:用户资料在/etc/passwd 文件中。

参 考 文 献

［1］吴学毅．Linux 基础教程［M］．北京：清华大学出版社，2005.

［2］鸟哥．鸟哥的 Linux 私房菜基础学习篇［M］．3 版．北京：人民邮电出版社，2010.

［3］杨志文．深入 Linux 构建与管理［M］．北京：中国青年出版社，2001.

［4］高俊峰．循序渐进 Linux 基础知识、服务器搭建、系统管理、性能调优、集群应用［M］．北京：人民邮电出版社，2009.

［5］IT 同路人．Linux 标准学习教程［M］．北京：人民邮电出版社，2008.

［6］Nemeth. Linux 系统管理技术手册［M］．张辉英，译．2 版．北京：人民邮电出版社，2008.

［7］Sarwar，Koretsky．LINUX 教程［M］．李善平，施韦，林欣，译．北京：清华大学出版社，2005.

ISBN 978-7-111-37309-4
ISBN 978-7-89433-316-2（光盘）
策划编辑：林春泉
封面设计：路恩中



零起点学自动化技术丛书

书名	书号
◎ 零起点学维修电工技术	36361
◎ 零起点学西门子变频器应用	36363
◎ 零起点学Linux系统管理	37309
◎ 零起点学Proteus单片机仿真技术	36904
◎ 零起点学西门子S7—200 PLC	
◎ 零起点学西门子S7—300/400 PLC	

上架指导：工业自动化技术/嵌入式系统

ISBN 978-7-111-37309-4



地址：北京市百万庄大街22号
电话服务
社服务中心：(010)88361066
销售一部：(010)68326294
销售二部：(010)88379649
读者购书热线：(010)88379203
邮政编码：100037
网络服务
门户网：http://www.cmpbook.com
教材网：http://www.cmpedu.com
封面无防伪标均为盗版

定价：29.80元（含1DVD）